MODEL KNOX

Created by **AccuKnox**

Zero Trust LLM Security

Secure Your AI Models from
Advanced Attacks in the GenAI Age

Written by

Atharva Shah

Get a tour of ModelKnox's Features to Secure LLMs from
Development to Deployment along with checklists and
recommendations from Gartner, Databricks, OWASP, NIST, MITRE

Zero Trust LLM Security

Powered by:

Atharva Shah

FOREWORD

In the era of Generative AI, security cannot be an afterthought. ModelKnox, offers a comprehensive solution for securing your Large Language Models (LLMs) throughout their lifecycle.

Key Features of ModelKnox -

- End-to-end LLM pipeline security
- Real-time threat detection and remediation
- Compliance with industry standards (OWASP, NIST, MITRE)
- Intuitive dashboard for holistic security posture management
- Securing CUDA workloads (NVIDIA) for enhanced performance and security

Here's what you can expect from ModelKnox -

- Mitigate risks associated with AI deployments
- Ensure data privacy and regulatory compliance
- Streamline DevSecOps and MLSecOps processes
- Stay ahead of emerging AI security threats

Don't let security concerns hold back your AI innovations. With ModelKnox, embrace the power of Large Language Models with confidence.

CONTENTS

1. INTRODUCTION

While earlier generations of AI technologies had significant promise and potential, current AI technologies deliver tangible business value, one that corporations, governments and individuals cannot ignore. Large language models (LLMs) got a lot of attention with the advent of ChatGPT, which popularized their use by businesses and individuals. AI technologies, that were relegated to the backwaters of niche corporate tasks or academic study, are now in the mainstream, thanks to advances in computer power, data availability, and tools such as Llama 2, ElevenLabs, and Midjourney.

The upshot is that the barrier to entry for GenAI has not only dropped precipitously but, as reflected in broad-based interest, it has also shown that solid plans for integrating and using AI in corporate processes are now a matter of urgency.

Gartner aptly opines in their most recent report:

> "AI agents must be protected like autonomous machines, across their tasks, interactions and communications" - Gartner, June 2024

Artificial intelligence (AI) is a multidisciplinary term covering numerous computer science subspecialties that make it feasible for robots to perform things that have historically required human intelligence. Two vital subspecialties of AI are machine learning (ML) and generative AI.

The primary goal is to create algorithms that can learn from data to make predictions or inferences. A type of machine learning often called GenAI is to learn a model to make new data, typically using large-scale language models. It is called a large language model if it is an AI system that uses massive natural language datasets to generate text that mimics that of a human.

The rapid development of GenAI technology has thrown enterprises into new terrain as they attempt to balance the benefits accrued with the need for extremely tight security measures. Fast progression in GenAI provides ways for adversaries to improve their methods of attack; this means that a dual challenge arises: the challenge for defense and the escalation of threats. Businesses use AI for controlled detection and response applications, email spam filtering, SIEM for behavioral analytics, and most HR aspects.

ModelKnox aims to provide security for the LLM application pipeline, right from development to runtime. As benefits, threat vectors and challenges of ethical and reliable AI evolve, ModelKnox will continue to advance and evolve to support you in protecting your LLM workloads.

ModelKnox is for tech, cybersecurity, privacy, compliance, legal, DevSecOps, MLSecOps, leaders to help them deliver highly value-added AI capability to business units so they derive strategic value. It helps such leaders leverage AI for company success, keep ahead in the space as AI regulations and requirements continue to be in flux, and at the same time curtail the perils that can come from increased attack surface from the increased adoption of AI/LLM Models.

ModelKnox relies on critical areas and responsibilities necessary for the organization's security in developing an LLM strategy. LLM application teams will benefit from this by enhancing the existing defense strategies and building new ones to mitigate the new threats that this state-of-the-art technology poses.

1. Views of Top Analysts on Gen AI Security

The Gartner document, titled "Emerging Tech: Top 4 Security Risks of GenAI" (G00785940), published on August 10, 2023, discusses the emerging security risks associated with generative AI (GenAI) technologies, particularly large language models (LLMs) and chat interfaces. The document identifies four major risks: privacy and data security, enhanced attack efficiency, misinformation, and fraud and identity risks.

"GenAI is expected to be fully capable of enhancing the efficiency of cyberattacks, which will challenge existing paradigms and security tooling in a variety of ways, including by providing in-depth knowledge support of known threat actor exploits and intelligence, Enabling potential future use of autonomous bots with in-depth penetration testing skills, Enhancing the ability for threat actors to bypass and evade enterprise security controls, Generating attacks or malware with simple text-based instructions via chatbot prompts and prompt injections" (Gartner, 2023).
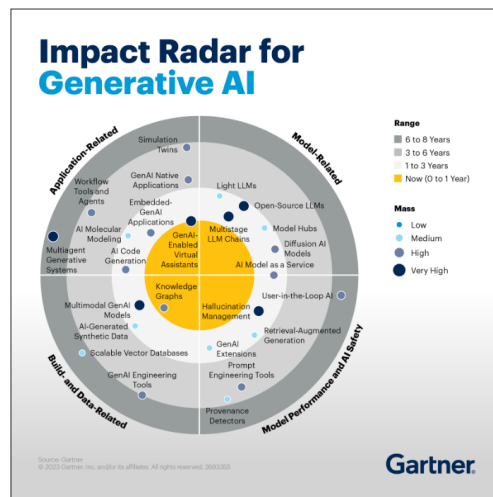
Other leading researchers have shared authoritative views on the importance of

securing AI LLM Models and the perils associated with cavalier use of un-protected AI assets. Here are a few examples:
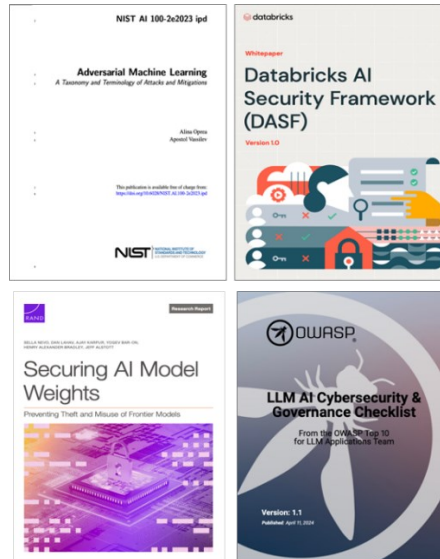
The following is DataBricks' view of 35 risks across AI components:

Raw Data	
1.1	Insufficient access controls
1.2	Missing data classification
1.3	Poor data quality
1.4	Ineffective storage and encryption
1.5	Lack of data versioning
1.6	Insufficient data lineage
1.7	Lack of data trustworthiness
1.8	Legality of data
1.9	Stale data
1.10	Lack of data access logs

Data Prep	
2.1	Pre-processing integrity
2.2	Feature Manipulation
2.3	Raw Data criteria
2.4	Adversarial partitions

Data Sets	
3.1	Data poisoning
3.2	Ineffective storage and encryption
3.3	Label Flipping

Governance	
4.1	Lack of traceability and transparency of model assets
4.2	Lack of end to end ML lifecycle

Algorithms	
5.1	Lack of tracking and reproducibility of experiments
5.2	Model drift
5.3	Hyperparameters stealing
5.4	Malicious libraries

Evaluation	
6.1	Evaluating data poisoning
6.2	Insufficient evaluation data

Model	
7.1	Backdoor Machine learning, Trojaned Model
7.2	Model assets leak
7.3	ML Supply Chain Vulnerabilities
7.4	Source code control attack

Model Management	
8.1	Model attribution
8.2	Model theft
8.3	Model lifecycle without HITL
8.4	Model Inversion

Model Serving	
9.1	Prompt Injection
9.2	Model Inversion
9.3	Model breakout
9.4	Looped input
9.5	Infer training data membership
9.6	Discover ML Model Ontology
9.7	Denial of Service
9.8	LLM Hallucinations
9.9	Input Resource Control
9.10	Accidental exposure of unauthorized data to models

Model Serving and Inference Response	
10.1	Lack of audit and monitoring inference quality
10.2	Output manipulation
10.3	Discover ML Model Ontology
10.4	Discover ML Model Family
10.5	Blackbox attacks

Platform	
12.1	Lack of vulnerability management
12.2	Lack of penetration testing and bug bounty
12.3	Lack of incident response
12.4	Unauthorized privileged access
12.5	Poor SDLC
12.6	Lack of compliance

1.1. Privacy and Data Security Risks

As per Gartner analysts, "The lack of data anonymization techniques, if not used

sufficiently and/or data is shared with third parties and with API authorization permissions management can lead to the potential data leak, risk or breach".

It also warns about the risk of "inference attacks" where advanced users can reverse-engineer sensitive information from the training data, and adversarial attacks that can bypass built-in safety measures through techniques like "jailbreaking" and "prompt injection."

1.2. Enhanced Attack Efficiency

"LLMs are increasing the availability and variety of attacks by enabling would-be attackers with malicious code development through easy-to-use prompts" (Gartner, 2023). It further states that "GenAI is expected to enhance the efficiency of future cyberattacks by enabling lower-level malicious actors with potentially novel and more advanced attacks"

The concept of smart malware leverages LLMs as autonomous agents to drive attack strategies, permutate attacks, and enable self-iteration and strategy development. They predict that by 2025, autonomous agents will drive advanced cyberattacks that give rise to 'smart malware,' pushing providers to offer innovations that address unique LLM and GenAI risks and threats

2. Recommendations from Gartner

The adoption of GenAI technologies in enterprises brings significant business value but also introduces new security risks, such as prompts and APIs.

Two trends are discussed: embedded GenAI applications and AI agents.

The trend of embedding external language models (LLMs) in GenAI applications raises security concerns, such as adversarial prompt attacks. Vendors must secure communication
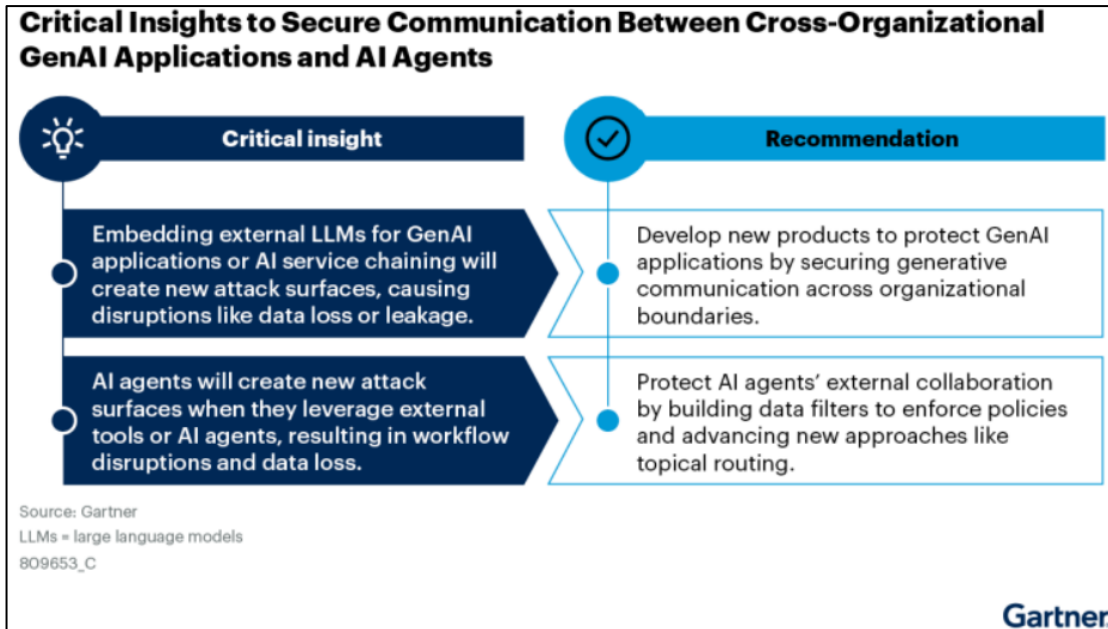
paths across enterprise boundaries. However, sending enterprise data to third parties via prompt engineering raises data leakage risks and breaks traditional security rules.
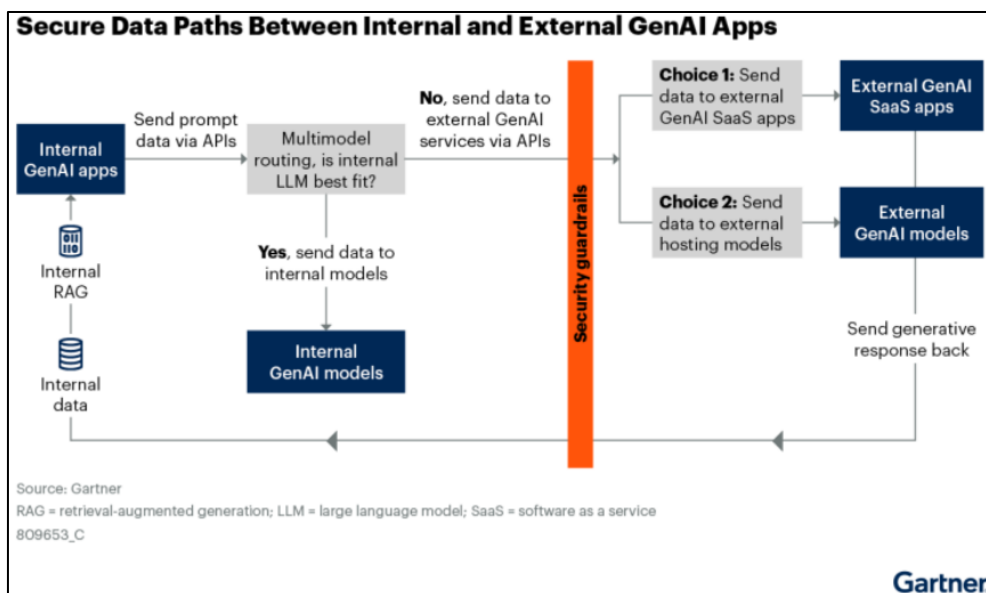
Gartner's 2023 AI in the Enterprise Survey shows the most common way to fulfill generative AI (GenAI) use cases is to embed large language models (LLMs) into existing applications.
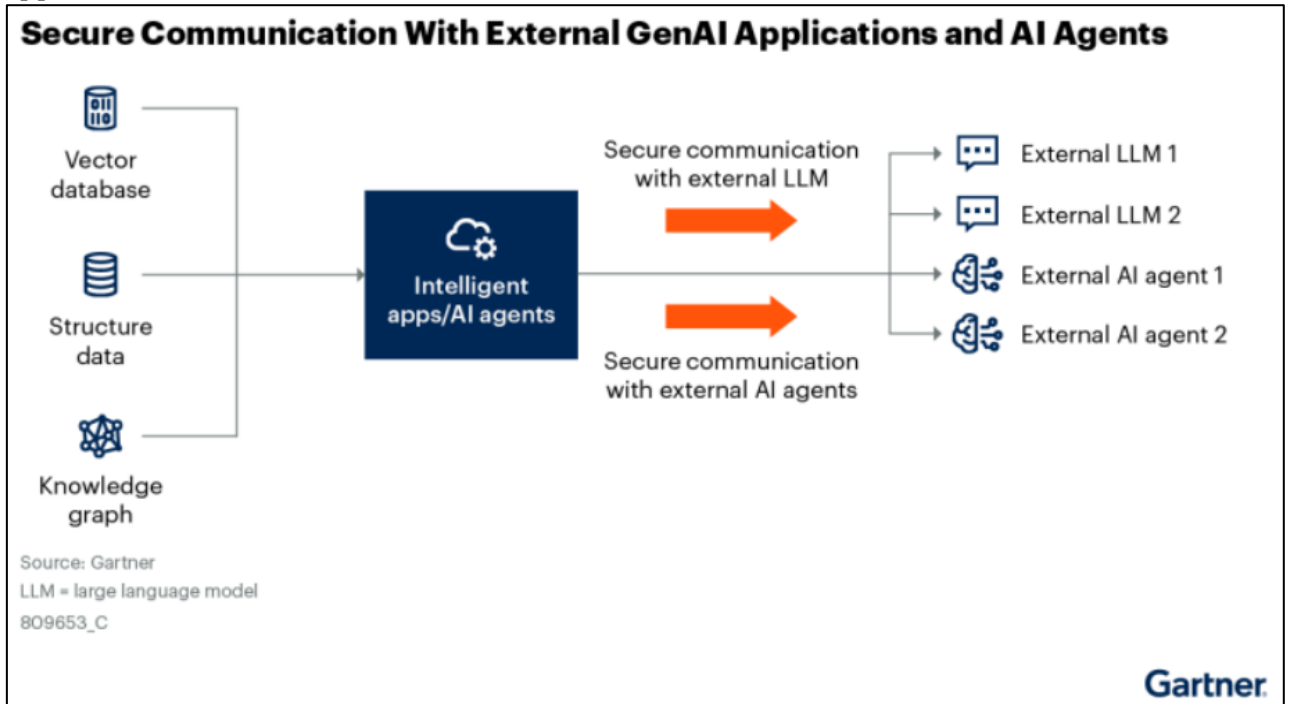
The increasing adoption of AI agents for automation raises security concerns due to uncontrolled data exchange with external parties. This trend creates business opportunities for vendors to secure communication paths across organizational boundaries. AI agents are autonomous or semi-autonomous software entities that use AI techniques to perceive, make decisions, take actions, and achieve goals in their digital or physical environments. Cross-organization AI agent workflows and data paths create new attack surfaces, such as APIs and uncontrolled data exchange, which should be protected before massive adoption.

Critical Insights to Secure Communication Between Cross-Organizational GenAI Applications and AI Agents

Critical insight	Recommendation
Embedding external LLMs for GenAI applications or AI service chaining will create new attack surfaces, causing disruptions like data loss or leakage.	Develop new products to protect GenAI applications by securing generative communication across organizational boundaries.
AI agents will create new attack surfaces when they leverage external tools or AI agents, resulting in workflow disruptions and data loss.	Protect AI agents' external collaboration by building data filters to enforce policies and advancing new approaches like topical routing.

Source: Gartner
LLMs = large language models
809653_C

Gartner

Externally hosted Language Modeling (LLMs) are being increasingly used for GenAI applications, presenting new security opportunities. However, this also raises concerns about prompt filtering, data security, and generative responses. Two common deployment styles involve enterprises without internal self-managed LLMs embedding their GenAI applications with external LLMs or implementing GenAI service chaining to route requests to external LLMs. Both scenarios involve common risks in prompting and data security, necessitating the implementation of security guardrails to intercept cross-organizational communication, apply security protocols to the data path, and perform filtering of prompts or data

Secure Data Paths Between Internal and External GenAI Apps

Internal GenAI apps → Send prompt data via APIs → Multimodel routing, is internal LLM best fit?

Internal RAG

Internal data

No, send data to external GenAI services via APIs → Security guardrails

Yes, send data to internal models → Internal GenAI models

Choice 1: Send data to external GenAI SaaS apps → External GenAI SaaS apps

Choice 2: Send data to external hosting models → External GenAI models

Send generative response back

Source: Gartner
RAG = retrieval-augmented generation; LLM = large language model; SaaS = software as a service
809653_C

Gartner

AccuKnox ModelKnox - Zero Trust LLM Security

AI agents are designed to act autonomously and proactively.

They learn and adapt by interacting with external tools and agents outside the organization's network. This collaboration can introduce risks, such as exposing new security attack surfaces that cause business disruption and asset damage. These agents often execute cross-organizational tasks, leveraging external tools and collaborating with external agents. As LLM-based AI agents become more popular, pool-to-pool agent communications are expected to become a new way to collaborate across B2B boundaries and split traffic between agent pools.

The need for efficiency in agent-to-agent communications will likely drive a greater emphasis on security elements. Agentic standards are nascent and not yet driven by outside developments beyond basic generative AI laws, such as the EU AI Act.

> "AI agents must be protected like autonomous machines, across their tasks, interactions, and communications." Gartner (June 2024)

How AI Agents Work

Source: Gartner
LLMs = large language models
8O9653_C
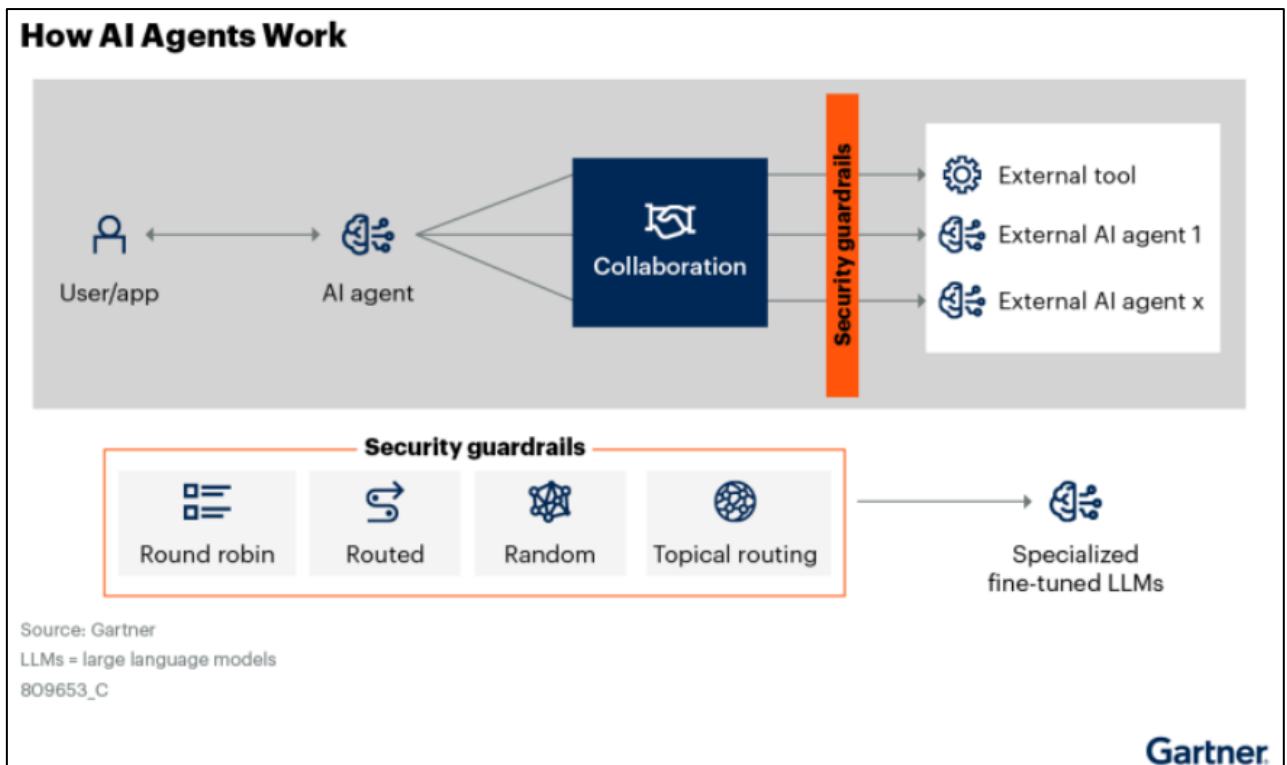
Gartner

Sample Security Risks and Ways to Mitigate Them (Source: Gartner)

Security Concerns	Recommended Features	Customer Outcome
Sensitive data leakage	Data masking, morphing or tokenization	Enhanced privacy protection

AccuKnox ModelKnox - Zero Trust LLM Security

Security Concerns	Recommended Features	Customer Outcome
Data loss	Data loss prevention (DLP)	Protection of enterprise data for confidentiality and integrity
Vulnerable APIs	API security solution including natural language analytics and filter for API payload	Usage restricted to approved external APIs; Customers have the ability to analyze text context inside API payload to avoid sensitive data being sent out
Malware, ransomware and other threats	Advanced threat detection and response tools	Prevention of threat attacks to external models and GenAI applications
Model safety/prompt filtering	Input/output filters, topical filters by AI agent or user role, binding of interactions to specific roles	Ability to control prompt data as input and model response as output
Shadow AI	Model use visibility, routing functions for specific topics or to approve or deny access	Ability to send prompt request only to the approved LLMs, restricting access to others
API authentication	Enforcement of geo-location, IP address or authorized hosts/apps for APIs	Prevention of the misuse of LLM APIs by third parties or through stolen credentials

Sample Attack Surfaces for Cross-Organization AI Agents (Source: Gartner)

Attack Surfaces	Security Concerns	Recommended Features for Product Roadmap
Risks arise at APIs and plug-ins when AI agents leverage external tools.	Data loss/leakage, API vulnerability, malware and ransomware attacks at the AI communication path	API security, API authentication, data masking, advanced threat detection and response tools including network detection and response

Attack Surfaces	Security Concerns	Recommended Features for Product Roadmap
		(NDR)/extended detection and response (XDR) products
Risks occur when internal AI agents communicate with external ones.	All concerns listed in Table 1 except model safety/prompt filtering and shadow AI; There are few or no security considerations in AI agent communication protocols.	API security, API authentication, data masking, morphing or tokenization, advanced threat detection and response tools; Design filters and define standards in AI agent communication protocols; Apply role bindings and other methods to guardrail LLM sessions, to control topics, and to enforce policy or access control

- Security providers should secure generative communication across organizational boundaries and deploy gateway products on their data paths to include features like data masking and API payload filters.
- Security products should build data filters to enforce AI agents' external communications, participating in security standard designs for agentic communication and advancing new approaches like topical enforcement.
- Platform should have methods for filtering and enforcing AI agents' communications that can classify topics, enforce content restrictions, filter complex data types and potentially move toward a more knowledge-aware enforcement era.
- Participate in the development of new AI agent standards to address new challenges in agentic communications and take an active role in defining industry standards to ensure the security of these communications.
- Advance the technology your organization uses in areas such as topical filtering, role enforcement, contextualized AI agent behavior and sensory agent's anomaly detection, with potential use cases to model the behavior of AI agents over time.

3. LLM Threat Vectors

OWASP [Open Web Application Security Project] is a nonprofit organization that aims

to improve software security and help protect web applications from cyber attacks. OWASP was founded in 2001 and has thousands of members and hundreds of chapters worldwide. OWASP has delivered its flagship report OWASP Top 10 on a number of areas (Cloud, API, End Point, etc.) that involve Security/Threat exposure. They recently released OWASP Top 10 for LLMs. The following section provides additional detail:

3.1. OWASP Top 10
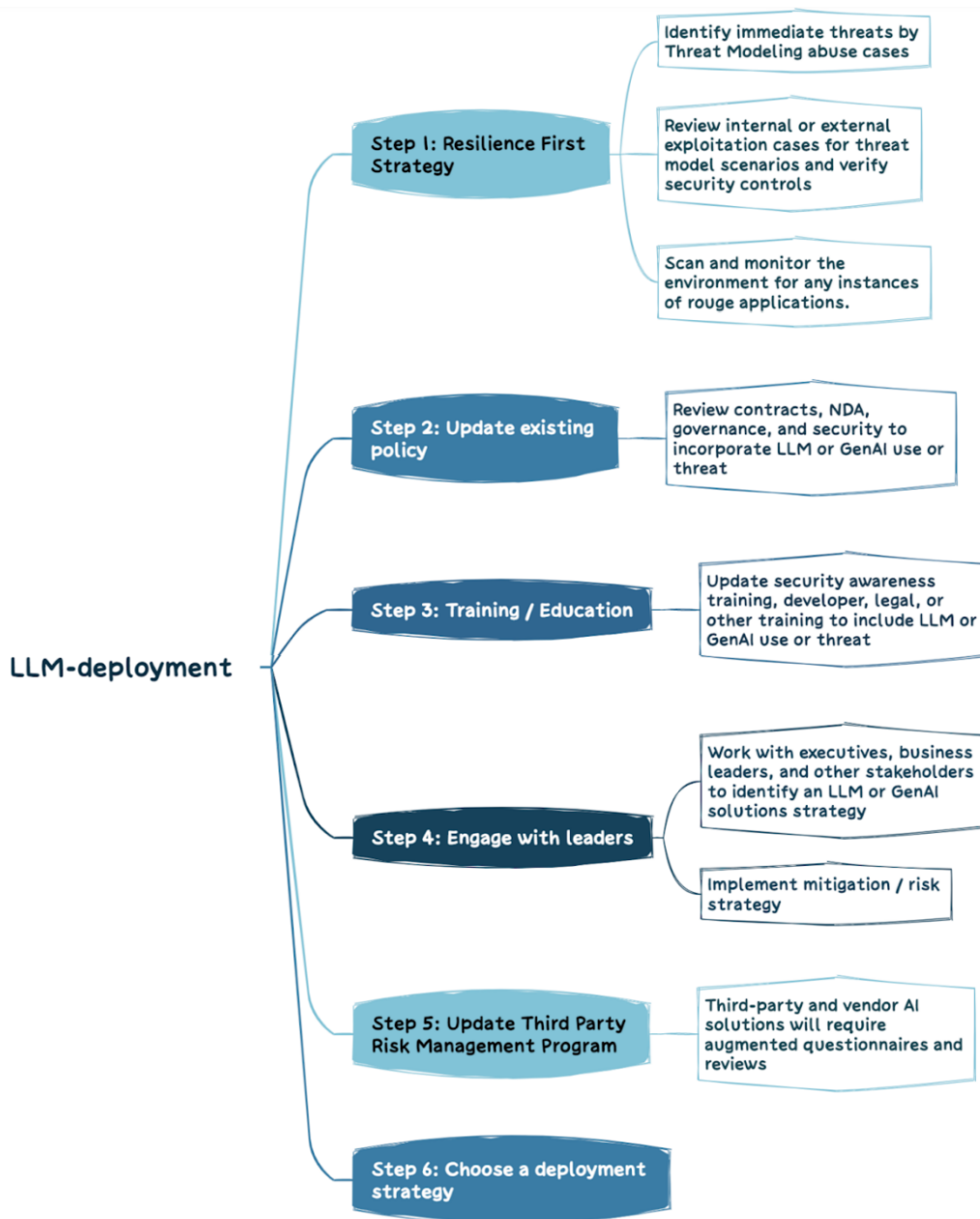
Source: OWASP LLM AI, Security, and Governance Checklist

AccuKnox ModelKnox - Zero Trust LLM Security

LLM01 - Prompt Injection

Prompt injection occurs when an attacker manipulates LLM inputs to execute unintended actions, potentially leading to data exfiltration, privilege escalation, or unauthorized access. This vulnerability arises from the lack of input validation and the LLM's inability to distinguish between benign and malicious prompts.

Common Examples of Vulnerability

1. Insufficient input validation allows attackers to inject malicious prompts that override the original system prompt.

2. Inadequate access controls enable attackers to manipulate external inputs fed into the LLM, leading to a "confused deputy" problem.

Example Attack Scenarios

1. An attacker injects a malicious prompt that instructs the LLM to ignore system prompts and execute commands that return sensitive information or perform unauthorized actions.

2. An attacker embeds a prompt injection in a webpage, which, when summarized by the LLM, causes it to exfiltrate sensitive user information via JavaScript or Markdown.

Prevention and Mitigation Strategies

1. Implement strict input validation and sanitization techniques to prevent malicious prompts from being processed by the LLM. Use a combination of allowlists, denylists, and regular expressions to filter out potentially harmful characters and sequences.

2. Enforce the principle of least privilege by limiting the LLM's access to sensitive data and functionality. Use granular access controls and API tokens to restrict the LLM's permissions to the minimum required for its intended purpose.

3. Regularly monitor and audit LLM inputs and outputs to detect anomalies and potential prompt injection attempts. Implement automated detection mechanisms that flag suspicious patterns or deviations from expected behavior.

4. Conduct regular penetration testing and red team exercises to identify and remediate prompt injection vulnerabilities proactively. Use a combination of manual testing and automated tools to simulate real-world attack scenarios and assess the LLM's resilience.

LLM02 - Insecure Output Handling

Insecure output handling occurs when LLM-generated content is passed downstream without proper validation, sanitization, or encoding. This can lead to various security risks, such as cross-site scripting (XSS), server-side request forgery (SSRF), and remote code

execution (RCE), depending on how the output is processed and used.

Common Examples of Vulnerability

1. LLM output is directly entered into a system shell or eval function without proper sanitization, leading to RCE.

2. JavaScript or Markdown generated by the LLM is returned to the user without proper encoding, resulting in XSS vulnerabilities.

Example Attack Scenarios

1. An attacker crafts a malicious prompt that causes the LLM to generate JavaScript code containing an XSS payload. When the output is rendered in a web application without proper sanitization, the attacker can steal sensitive user information or perform unauthorized actions.

2. An LLM plugin generates SQL queries based on user input without proper parameterization or escaping. An attacker exploits this vulnerability by injecting malicious SQL statements, leading to database compromise or data exfiltration.

Prevention and Mitigation Strategies

1. Implement comprehensive output validation and sanitization mechanisms to ensure that LLM-generated content is safe before being passed downstream. Use well-established libraries and frameworks that provide built-in security features, such as parameterized queries and context-aware encoding.

2. Follow the OWASP Application Security Verification Standard (ASVS) guidelines for input validation, sanitization, and output encoding. Ensure that all user-supplied input is treated as untrusted and properly validated before being processed by the LLM or any downstream components.

3. Use secure coding practices and perform regular code reviews to identify and remediate insecure output handling vulnerabilities. Pay special attention to areas where LLM output is used in dynamic contexts, such as SQL queries, shell commands, or HTML rendering.

4. Implement a Content Security Policy (CSP) to mitigate the impact of XSS vulnerabilities by restricting the execution of inline scripts and limiting the sources of executable content.

5. Utilize security testing techniques, such as static application security testing (SAST) and dynamic application security testing (DAST), to identify insecure output handling issues during the development and testing phases.

6. Regularly update and patch LLM components and downstream systems to address known vulnerabilities and security weaknesses. Keep track of security advisories and

incorporate relevant security fixes promptly.

LLM03 - Training Data Poisoning

Training data poisoning is the manipulation of pre-training data, fine-tuning data, or embedding processes to introduce vulnerabilities, backdoors, or biases into the LLM. This can lead to compromised model security, degraded performance, or the generation of harmful or misleading outputs.

Common Examples of Vulnerability

1. Insufficient data validation and sanitization allow attackers to inject malicious samples into the training dataset, corrupting the LLM's learned patterns.

2. Inadequate access controls and authentication mechanisms enable unauthorized modification of training data or model artifacts.

Example Attack Scenarios

1. An attacker injects a large number of carefully crafted samples into the LLM's fine-tuning dataset, causing the model to learn unintended patterns or behaviors. This can lead to the generation of biased, misleading, or harmful outputs.

2. A malicious insider with access to the training pipeline manipulates the embedding process by altering word vectors or introducing backdoors. This allows the attacker to control the LLM's behavior or exfiltrate sensitive information.

Prevention and Mitigation Strategies

1. Enforce strict access controls and authentication mechanisms for the training pipeline and model artifacts. Use role-based access control (RBAC) and multi-factor authentication (MFA) to limit access to authorized personnel only.

2. Utilize secure storage and transmission methods for training data and model artifacts. Encrypt sensitive data at rest and in transit using strong encryption algorithms and secure protocols.

3. Implement version control and auditing mechanisms for training data and model artifacts. Maintain a complete history of changes and regularly review logs for suspicious activities or unauthorized modifications.

4. Conduct regular security audits and penetration testing of the training pipeline and model hosting environment. Engage third-party security experts to assess the system's resilience against data poisoning attacks and provide recommendations for improvement.

5. Implement anomaly detection and outlier analysis techniques to identify and flag suspicious patterns or deviations in the training data. Use machine learning algorithms

to detect and alert on potential poisoning attempts in real-time.

LLM04 - Model Denial of Service

Model Denial of Service (DoS) occurs when an attacker overwhelms an LLM with crafted inputs that consume excessive computational resources, leading to degraded performance, increased latency, or complete unavailability of the service. This can impact the LLM's responsiveness and cause financial losses due to increased resource consumption.

Common Examples of Vulnerability

1. Insufficient input validation and rate limiting allow attackers to submit a large number of resource-intensive queries, exhausting the LLM's processing capacity.

2. Inadequate monitoring and detection mechanisms fail to identify and mitigate DoS attacks in real-time, leading to prolonged service disruptions.

Example Attack Scenarios

1. An attacker submits a large number of queries containing complex linguistic patterns, nested references, or recursive structures that trigger worst-case computational complexity in the LLM. This causes the model to consume excessive memory and CPU resources, leading to degraded performance for legitimate users.

2. An attacker exploits a vulnerability in the LLM's tokenization or sequence processing pipeline to submit specially crafted inputs that cause the model to enter an infinite loop or exhaust available resources. This results in a complete denial of service, rendering the LLM unavailable to other users.

Prevention and Mitigation Strategies

1. Implement robust input validation and sanitization mechanisms to filter out malicious or resource-intensive queries. Use regular expressions, character limits, and other techniques to enforce strict input constraints and prevent abuse.

2. Enforce rate limiting and throttling mechanisms to control the number of requests that can be submitted by a single user or IP address within a given time frame. Use sliding windows or token bucket algorithms to balance resource utilization and prevent DoS attacks.

3. Implement resource isolation and containerization techniques to limit the impact of resource-intensive queries on other users. Use virtualization, containerization, or serverless architectures to allocate dedicated resources for each request and prevent resource exhaustion.

4. Implement monitoring and alerting mechanisms to detect unusual spikes in resource utilization or request volume. Use log analysis, metrics collection, and anomaly detection

algorithms to identify potential DoS attacks in real-time and trigger automated mitigation measures.

LLM05 - Supply Chain Vulnerabilities

Supply chain vulnerabilities in LLMs refer to weaknesses introduced through the use of compromised or malicious components, such as pre-trained models, training datasets, or third-party libraries. These vulnerabilities can lead to data leakage, model corruption, or the introduction of backdoors and trojans into the LLM system.

Common Examples of Vulnerability

1. Incorporation of pre-trained models or datasets from untrusted sources without proper validation and scrutiny, leading to the introduction of malicious artifacts or biases.

2. Dependence on outdated or vulnerable third-party libraries with known security issues, exposing the LLM system to potential exploitation.

Example Attack Scenarios

1. An attacker injects a backdoored pre-trained model into the LLM supply chain, allowing them to control the model's behavior or exfiltrate sensitive information processed by the LLM.

2. A malicious actor compromises a widely used dataset repository and inserts poisoned samples into popular training datasets. LLM systems that rely on these datasets unknowingly incorporate the malicious data, leading to corrupted model outputs or unauthorized access.

Prevention and Mitigation Strategies

1. Implement a robust vendor risk management program to assess and mitigate risks associated with third-party components used in the LLM system. Conduct thorough due diligence on suppliers, including security assessments, code reviews, and background checks.

2. Implement secure coding practices and follow the OWASP Secure Coding Guidelines to minimize the risk of introducing vulnerabilities during the development and integration of LLM components. Conduct regular code reviews and security audits to identify and remediate potential weaknesses.

3. Utilize static application security testing (SAST) and dynamic application security testing (DAST) tools to automatically scan the LLM codebase and dependencies for known vulnerabilities. Integrate these tools into the continuous integration and continuous deployment (CI/CD) pipeline to catch issues early in the development process.

4. Implement a vulnerability management program to continuously monitor for new

vulnerabilities in the LLM supply chain components. Subscribe to security advisories, mailing lists, and vulnerability databases to stay informed about newly discovered issues and patches.

5. Enforce strict access controls and authentication mechanisms for the LLM development and deployment environments. Use role-based access control (RBAC), multi-factor authentication (MFA), and secure key management practices to prevent unauthorized access to sensitive components and data.

LLM06 - Sensitive Information Disclosure

LLMs may inadvertently reveal sensitive information, proprietary algorithms, or confidential data through their generated outputs. This can lead to data breaches, privacy violations, and intellectual property theft.

Common Examples of Vulnerability

1. Insufficient output filtering allows the LLM to include sensitive data in its responses.
2. Inadequate access controls enable unauthorized users to query the LLM and obtain confidential information.

Example Attack Scenarios

1. An attacker crafts specific prompts that trick the LLM into revealing sensitive data, such as user PII or internal system details.
2. A malicious user exploits weak access controls to query the LLM and obtain proprietary algorithms or trade secrets.

Prevention and Mitigation Strategies

1. Implement robust output filtering and sanitization techniques to remove sensitive information from LLM responses.
2. Enforce granular access controls and authentication mechanisms to limit LLM access to authorized users only.
3. Regularly audit and monitor LLM inputs and outputs to detect potential data leakage.
4. Provide clear guidelines and training for users on how to interact with the LLM securely.

LLM07 - Insecure Plugin Design

Poorly designed LLM plugins can introduce vulnerabilities that allow attackers to execute malicious code, escalate privileges, or access unauthorized resources. These vulnerabilities often stem from inadequate input validation and access control mechanisms within the plugins.

AccuKnox ModelKnox - Zero Trust LLM Security

Common Examples of Vulnerability

1. Plugins that accept unsanitized user input, enabling command injection or cross-site scripting attacks.

2. Plugins with excessive privileges that can be exploited to access sensitive data or perform unauthorized actions.

Example Attack Scenarios

1. An attacker crafts a malicious input that exploits a plugin's lack of input validation, allowing them to execute arbitrary code or commands.

2. A plugin with excessive permissions is compromised, enabling the attacker to access sensitive resources or perform unauthorized actions.

Prevention and Mitigation Strategies

1. Follow secure coding practices and perform thorough input validation and sanitization within LLM plugins.

2. Implement the principle of least privilege, granting plugins only the necessary permissions to perform their intended functions.

3. Conduct regular security audits and penetration testing of LLM plugins to identify and address vulnerabilities.

4. Implement secure communication channels and authentication mechanisms between the LLM and its plugins.

LLM08 - Excessive Agency

LLMs may be granted excessive agency, allowing them to perform actions or access resources beyond their intended scope. This can lead to unintended consequences, such as unauthorized data modification, system manipulation, or privacy violations.

Common Examples of Vulnerability

1. LLMs with unconstrained access to external APIs or services, enabling them to perform unauthorized actions.

2. Inadequate monitoring and control mechanisms to detect and prevent excessive agency.

Example Attack Scenarios

1. An LLM with excessive agency interacts with an external API, inadvertently modifying or deleting sensitive data.

2. A malicious actor exploits an LLM's excessive agency to manipulate system behavior or access unauthorized resources.

AccuKnox ModelKnox - Zero Trust LLM Security

Prevention and Mitigation Strategies

1. Implement strict boundaries and constraints on the actions an LLM can perform and the resources it can access.

2. Utilize secure API gateways and access control mechanisms to enforce LLM permissions and limit excessive agency.

3. Implement monitoring and auditing mechanisms to detect and alert on unusual LLM behavior.

4. Regularly review and update LLM policies and permissions to ensure they align with the intended functionality.

LLM09 - Overreliance

Over Reliance on LLMs can lead to the acceptance of inaccurate, biased, or misleading information without proper verification. This can result in flawed decision-making, propagation of misinformation, and reputational damage.

Common Examples of Vulnerability

1. Insufficient validation and fact-checking of LLM-generated content.

2. Lack of human oversight and critical analysis of LLM outputs.

Example Attack Scenarios

1. An attacker manipulates an LLM to generate misleading or false information, which is then relied upon by users or downstream systems.

2. Over Reliance on LLM-generated content leads to the spread of biased or inaccurate information, causing reputational harm.

Prevention and Mitigation Strategies

1. Implement robust validation and fact-checking mechanisms to verify the accuracy and reliability of LLM-generated content.

2. Encourage human oversight and critical analysis of LLM outputs, especially for high-stakes decisions.

3. Provide clear disclaimers and educate users about the limitations and potential biases of LLMs.

4. Regularly monitor and audit LLM-generated content for accuracy, bias, and potential misuse.

LLM10 - Model Theft

Model theft involves the unauthorized access, extraction, or replication of an LLM's underlying architecture, parameters, or training data. This can lead to intellectual property infringement, competitive disadvantage, and the potential misuse of the stolen model.

Common Examples of Vulnerability

1. Inadequate access controls and authentication mechanisms protecting the LLM and its associated resources.

2. Insufficient monitoring and detection capabilities to identify and respond to model theft attempts.

3. Example Attack Scenarios

4. An attacker gains unauthorized access to the LLM's storage or deployment environment and extracts the model's parameters or architecture.

5. A malicious insider with privileged access steals the LLM's training data or model artifacts for personal gain or to share with competitors.

Prevention and Mitigation Strategies

1. Implement strong access controls, authentication, and authorization mechanisms to protect the LLM and its associated resources.

2. Encrypt sensitive model artifacts and training data at rest and in transit.

3. Employ secure storage and transmission protocols to prevent unauthorized access or interception of model data.

4. Implement robust monitoring, logging, and alerting mechanisms to detect and respond to suspicious activities or model theft attempts.

5. Conduct regular security audits and penetration testing to identify and address vulnerabilities in the LLM's infrastructure and deployment environment.

In summary, OWASP Top 10 LLM provides a very comprehensive set of vulnerabilities, associated risks and mitigation strategies

3.2. NIST

The rise of large language models (LLMs) has brought about unprecedented capabilities in natural language processing, but it has also introduced significant privacy concerns. As these models are trained on vast amounts of data, often including sensitive or personal information, they can become susceptible to various privacy attacks. The National Institute of Standards and Technology (NIST) has outlined several critical privacy risks associated with LLMs and provided guidelines for mitigating these threats.

AccuKnox ModelKnox - Zero Trust LLM Security

Type of Attack	Description
Data Reconstruction Attacks	• Exploit LLM knowledge to reconstruct training data. • NIST recommends implementing differential privacy during training. • Even with differential privacy, the risk of reconstructing rare data remains.
Memorization Attacks	• LLMs inadvertently disclose sensitive training data during inference. • Implement content filtering and redaction mechanisms. • LLMs may still leak sensitive data subtly or in coded ways.
Membership Inference Attacks	• Adversaries determine if data was in the training set. • NIST suggests differential privacy and secure computation. • Balancing privacy techniques with model performance is a challenge.
Model Extraction Attacks	• Steal or replicate model parameters/functionality. • Implement access control, encryption, watermarking. • Research ongoing on effectiveness against advanced techniques.
Property Inference Attacks	• Infer sensitive properties of training data. • Conduct privacy impact assessments, data minimization. • High-dimensional representations pose challenges in prevention.

Mitigations and Best Practices
- NIST emphasizes the importance of adopting a holistic approach to LLM security, involving a combination of technical, organizational, and policy-based measures.
- Key recommendations include:
  - Robust access controls and encryption for LLM models and data
  - Regular security assessments and penetration testing
  - Clear policies and procedures for LLM development, deployment, and use
  - Promoting privacy-preserving techniques, such as differential privacy and secure

  multi-party computation
- According to industry reports, only 28% of organizations have implemented end-to-end security measures for their LLM deployments, highlighting the need for increased awareness and adoption of best practices.

As LLMs continue to evolve and permeate various industries, addressing privacy concerns will be crucial for building trust and ensuring responsible adoption of these powerful technologies. ModelKnox solution is designed and developed with a strong emphasis on adhering to NIST's guidelines and recommendations for LLM security. By incorporating best practices such as robust access controls, encryption, privacy-preserving techniques, and holistic security assessments, ModelKnox provides a robust and trustworthy platform for securing LLM-based workloads. We also actively participate in ongoing research efforts, and collaborate with industry leaders to stay ahead of emerging privacy threats and contribute to the development of cutting-edge solutions for LLM security. By aligning with NIST recommendations ModelKnox leverages the power of LLMs to upkeep the highest levels of privacy protection.

3.3. MITRE

ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) is a globally accessible, living knowledge base of adversary tactics and techniques against AI-enabled systems based on real-world attack observations and realistic demonstrations from AI red teams and security groups. ModelKnox covers every single part of this comprehensive attack pipeline, providing robust security for LLMs throughout the entire lifecycle.

Source: Atlas Matrix

Reconnaissance: Involves gathering information about the target system or victim. The techniques include Search for Victim's Publicly Available Research Materials, Search for Publicly Available Adversarial Vulnerability Analysis, Search Victim-Owned Websites, Search Application Repositories, and Active Scanning.

AccuKnox ModelKnox - Zero Trust LLM Security

Resource Development: The attacker acquires or develops the necessary resources for carrying out the attack. The techniques include Acquire Public ML Artifacts, Obtain Capabilities, Develop Capabilities, Acquire Infrastructure, Publish Poisoned Datasets, and Poison Training Data.

Initial Access: The focus here is on gaining initial access to the target system or ML model. The techniques are ML Supply Chain Compromise, Valid Accounts, Exploit Public-Facing Application, LLM Prompt Injection, and Phishing.

ML Model Execution: Involves executing the ML model or interacting with it. The techniques are ML Model Inference API Access, ML-Enabled Product or Service, User Execution, Command and Scripting Interpreter, and Physical Environment Access.

Execution: The attacker executes the payload or the malicious code. The techniques are Poison Training Data, Backdoor ML Model, LLM Prompt Injection, LLM Plugin Compromise, and Full ML Model Access.

Persistence: The aim is to maintain a persistent presence in the target system or ML model. The techniques are Poison Training Data, Backdoor ML Model, LLM Prompt Injection, and LLM Plugin Compromise.

Privilege Escalation: Involves escalating privileges or gaining higher levels of access. The technique listed is LLM Prompt Injection.

Defense Evasion: Attempts are made to evade detection or bypass security controls. The techniques are Evade ML Model, Unsecured ML Credentials, and LLM Prompt Injection.

Credential Access: Goal is to obtain credentials or authentication materials. The technique listed is Discover ML Credentials.

Discovery: Discovering and mapping the target environment or ML infrastructure. The techniques are Discover ML Model Ontology, Discover ML Model Family, Discover ML Artifacts, LLM Meta Prompt Extraction, and Craft Adversarial Data.

Collection: Extracting data or artifacts from the target system or ML model. The techniques are Data from Information Repositories, Backdoor ML Model, and LLM Meta Prompt Extraction.

ML Attack Staging: Staging or preparing for the final attack. The technique listed is Create Proxy ML Model.

Exfiltration: Extracting or exfiltrating data or artifacts from the target system or ML model. The techniques are Exfiltration via ML Inference, Exfiltration via Cyber Means, Data from Local System, Verify Attack, LLM Meta Prompt Data, and LLM Data Leakage.

Impact: Represents the ultimate goal or impact of the attack. The techniques are Evade ML Model, Denial of ML Service, Spamming ML System with Chaff Data, Erode ML Model Integrity, Cost Harvesting, and External Harms.

3.4. Databricks AI Security Framework (DASF) Recommendations

In contrast to approaches that narrowly focus on securing models or endpoints, ModelKnox adopts a strategy aimed at mitigating cyber risks across all facets of AI systems. Recognizing that attackers often exploit simple tactics to compromise ML-driven systems, ModelKnox offers actionable defensive control recommendations derived from real-world evidence. Central to ModelKnox's effectiveness is its adaptability. The framework's recommendations are continuously updated to reflect emerging risks and the availability of additional controls. This dynamic approach ensures that organizations leveraging ModelKnox remain resilient in the face of evolving cyber threats. The development of ModelKnox involved a meticulous review of multiple risk management frameworks, recommendations, whitepapers, policies, and AI security acts. By incorporating insights from a diverse array of sources, ModelKnox aligns with industry best practices and regulatory requirements, providing organizations with a robust framework for securing their AI systems. It was built by integrating compliance with regulations, rules, and recommendations, ModelKnox empowers organizations to harness the transformative potential of AI while safeguarding against cyber threats with confidence.
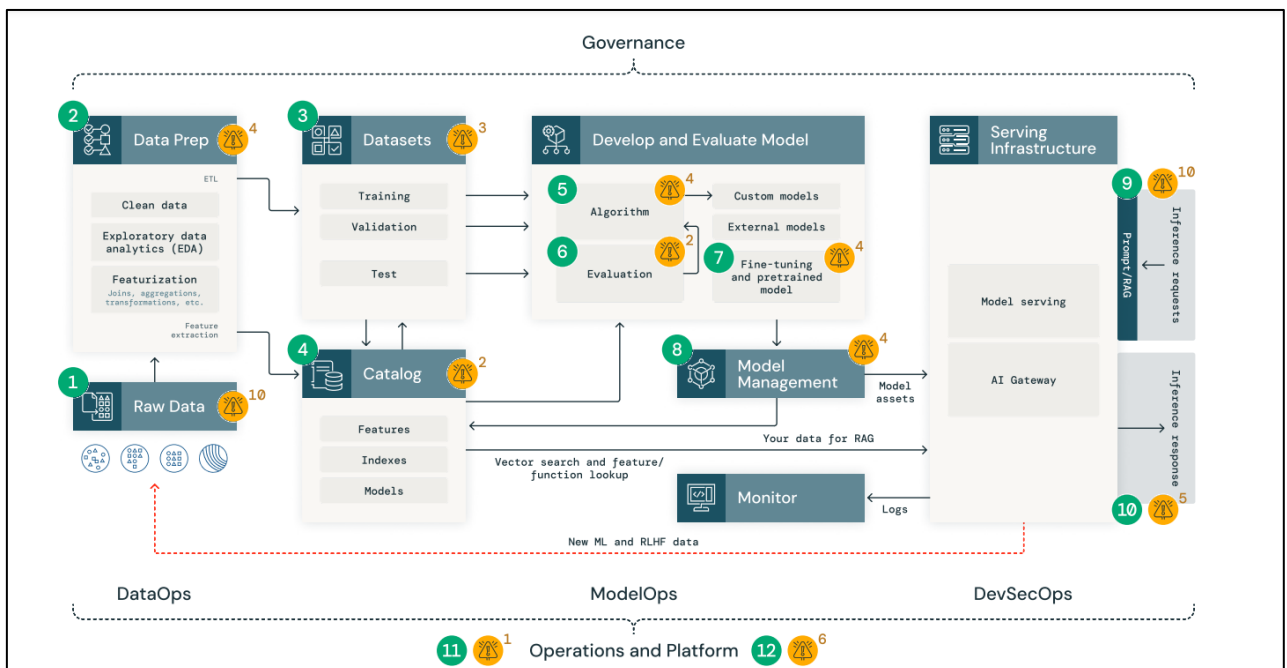
AccuKnox ModelKnox - Zero Trust LLM Security

Insufficient Access Controls

One of the fundamental pillars of data security within AI systems lies in the establishment of robust access controls. Without effective protocols governing who can access datasets and under what circumstances, organizations are left vulnerable to unauthorized access and potential data breaches. Proper access management is critical for safeguarding sensitive information and ensuring compliance with regulatory requirements.

Missing Data Classification

Failure to classify data based on its sensitivity, importance, and criticality poses a significant threat to AI systems. Without clear classification criteria, organizations struggle to implement appropriate security measures and governance policies tailored to the specific needs of different types of data. This lack of granularity can lead to inadequate protection of valuable assets and increased susceptibility to security incidents.

Poor Data Quality

The integrity, accuracy, and consistency of data are paramount for the reliability of analytics and decision-making processes in AI systems. Poor data quality compromises the foundation upon which AI models are built, leading to erroneous outcomes and potentially severe consequences for organizations. Addressing data quality issues requires comprehensive data governance frameworks and robust data validation mechanisms.

Ineffective Storage and Encryption

Secure data storage and encryption are essential components of a comprehensive data security strategy. Inadequate measures in these areas expose sensitive information to potential breaches and unauthorized access, posing significant risks to organizations. Implementing robust encryption protocols and ensuring secure storage practices are essential for protecting data both at rest and in transit.

Lack of Data Versioning

The ability to track and trace data to its original state is crucial for maintaining data integrity and facilitating recovery efforts in the event of data corruption or tampering. Without proper data versioning mechanisms in place, organizations face challenges in identifying and reverting to previous iterations of data, potentially leading to prolonged downtime and increased vulnerability to security threats.

Insufficient Data Lineage

Transparency and traceability in data usage are essential for ensuring compliance with regulations and auditing requirements. Insufficient data lineage mechanisms hinder organizations' ability to track the origin and transformation of data throughout its lifecycle, making it difficult to demonstrate accountability and enforce data governance policies effectively.

Lack of Data Trustworthiness

Validating data sources and implementing integrity checks are crucial for establishing trust in the data used for training AI models. Without mechanisms to verify the trustworthiness of data, organizations risk exposure to data tampering or poisoning, which can undermine the reliability and credibility of AI-driven insights and decisions.

Data Legal Concerns

Navigating intellectual property concerns and legal mandates surrounding data usage is critical for organizations leveraging AI systems. Compliance with regulations requiring the capability to delete specific data from machine learning systems is essential to avoid legal repercussions and ensure ethical data practices.

Stale Data

Outdated or inaccurate data can lead to delays in business processes and undermine the performance of AI systems. Organizations must implement processes to regularly review and update datasets to ensure they remain relevant and accurate for AI applications.

Lack of Data Access Logs

Auditing mechanisms are essential for organizations to monitor access to data and identify potential security threats. The lack of comprehensive data access logs leaves organizations unaware of their risk surface area, increasing the likelihood of data breaches and regulatory non-compliance. Implementing robust logging mechanisms is essential for maintaining visibility and accountability in data access activities.

3.5. RAND Recommendations for Securing Frontier AI Models

The RAND Corporation report focuses on improving security for frontier AI models, particularly large language models and multimodal models. It emphasizes protecting model weights, which are crucial to a model's intelligence and capabilities. Key points include:

1. Identification of 38 distinct attack vectors, with examples of real-world successful attacks.
2. Categorization of attacker capabilities, from opportunistic criminals to state actors.

3. Assessment of attack vector feasibility for different attacker categories.

4. Proposal of five security levels with corresponding benchmark security systems.

5. Urgent recommendations for frontier AI organizations:
   - Develop a comprehensive security plan
   - Centralize and limit access to model weights
   - Harden interfaces against weight exfiltration
   - Implement insider threat programs
   - Invest in defense-in-depth strategies
   - Engage in advanced third-party red-teaming
   - Incorporate confidential computing

6. Long-term security measures:
   - Physical bandwidth limitations
   - Development of specialized hardware for weight security
   - Establishment of isolated networks for training and research

Attack Category	Attack Vector
Running Unauthorized Code	• Exploiting vulnerabilities for which a patch exists (attacking non-updated software) • Exploiting reported but not (fully) patched vulnerabilities • Finding and exploiting individual zero-days • Direct access to zero-days at scale
Compromising Existing Credentials	• Social engineering • Password brute-forcing and cracking • Exploitation of exposed credentials • Expanding illegitimate access (e.g., escalating privileges)
Undermining the Access Control System Itself	• Encryption/authentication vulnerabilities (in the access control system) • Intentional backdoors in algorithms, protocols, or products (in the access control system) • Code vulnerabilities (in the access control system) • Access to secret material undermining a protocol
Bypassing Primary Security System Altogether	• Incorrect configuration or security policy implementation • Additional (less secure) copies of sensitive data • Alternative (less secure) authentication or access schemes
AI-Specific Attack	• Discovering existing vulnerabilities in the ML stack

Attack Category	Attack Vector
Vectors	• Intentional ML supply chain compromise • Prompt-triggered code execution • Model extraction • Model distillation
Nontrivial Access to Data or Networks	• Digital access to air-gapped networks • Side-channel attacks (including through leaked emanations; i.e., TEMPEST attacks) • Eavesdropping and wiretaps
Unauthorized Physical Access to Systems	• Direct physical access to sensitive systems • Malicious placement of portable devices • Physical access to devices in other locations
Supply Chain Attacks	• Services and equipment the organization uses • Code and infrastructure incorporated into the codebase • Vendors with access to information
Human Intelligence	• Bribes and cooperation • Extortion • Candidate placement • Organizational leverage attacks • Organizationally approved access

4. ModelKnox

4.1. Why Do 80% of Security Leaders Rank AI as a Critical Risk?

AI and LLMs are reshaping industries, driving automation, enhancing customer experiences, optimizing processes, and unlocking new business opportunities. However, this transformative progress also presents a new set of cybersecurity challenges that organizations must be prepared to tackle.

AccuKnox ModelKnox - Zero Trust LLM Security

| Jailbreaking | Prompt injection | Backdoor and data poisoning | Adversarial inputs | Insecure output handling |
| Data extraction and privacy | Data reconstruction | Denial of service | Watermarking and evasion | Model theft |

Some of the common issues associated with AI and LLM technologies include:

1. Discovery of LLM models: The presence of LLM models within an organization's infrastructure is often a blind spot for security teams. When left unchecked, this can introduce data security and privacy risks. Without proper oversight, LLMs may accidentally expose sensitive information, becoming vulnerable to attacks such as prompt injection or data leakage. Furthermore, the unauthorized or improper use of LLMs can lead to the generation of biased or inappropriate content.

2. Prompt injection attacks: These attacks involve injecting malicious inputs into the prompts provided to AI models, manipulating the model's output. This can lead to unintended consequences, such as the disclosure of sensitive information or the execution of harmful actions.

3. Sensitive information disclosure: LLMs, if improperly secured, can accidentally reveal sensitive data, including internal configurations, user data, or proprietary information. This often occurs due to insecure configurations, flawed application design, or failure to sanitize data properly.

4. Model theft: Also known as model extraction, this threat involves attackers duplicating a machine learning model without direct access to its parameters or training data. Attackers can use query-based techniques to reverse engineer the model, posing significant risks to intellectual property.

5. Data leakage: Unauthorized transmission of confidential data can occur through various means, including insecure handling practices or the AI's inadvertent inclusion of sensitive information in its responses.

6. Compliance and reputational risks: The misuse of AI and LLMs can result in compliance violations, especially concerning data protection regulations. Moreover, the generation of inappropriate or biased content by these models can cause

significant reputational harm to organizations.
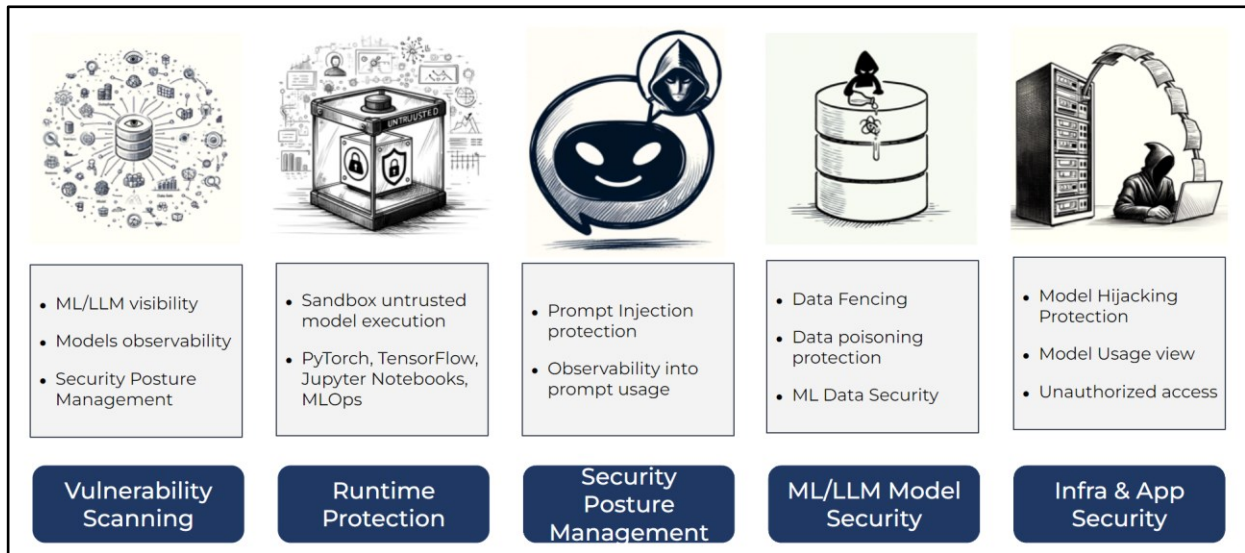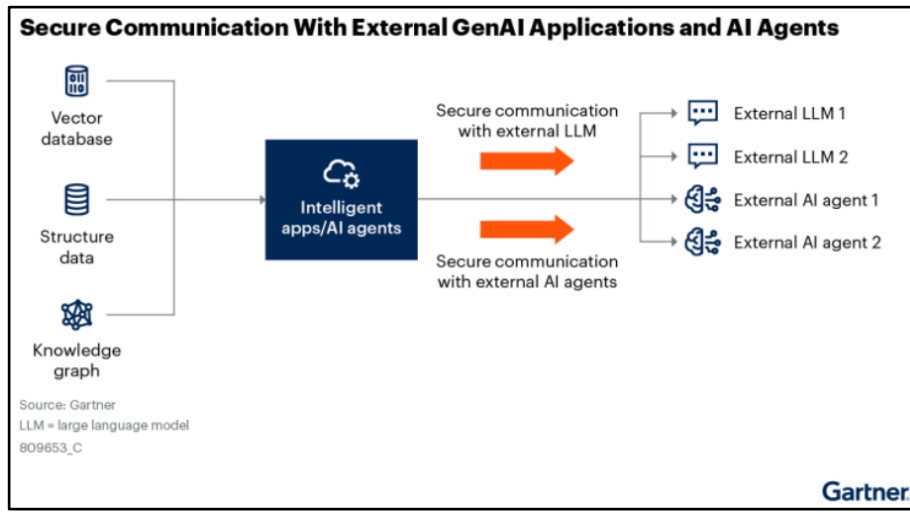
4.2. Understanding the AI/ML Pipeline Across Public Clouds

To effectively secure AI and LLM applications, it's crucial to understand the complex web of components and interactions within the AI/ML pipeline across major public cloud platforms, such as Google Cloud Platform (GCP), Microsoft Azure, and Amazon Web Services (AWS).

The typical AI/ML pipeline involves stages like data ingestion, model training, deployment, and inference. Each stage introduces potential security risks that need to be managed effectively.

Visualizing the AI/ML pipeline across these platforms is essential for understanding the

flow of data and identifying where vulnerabilities might arise. This graphical representation provides a comprehensive overview of the various components involved, enabling security teams to better assess and mitigate risks.

This is exactly what ModelKnox graph view does.
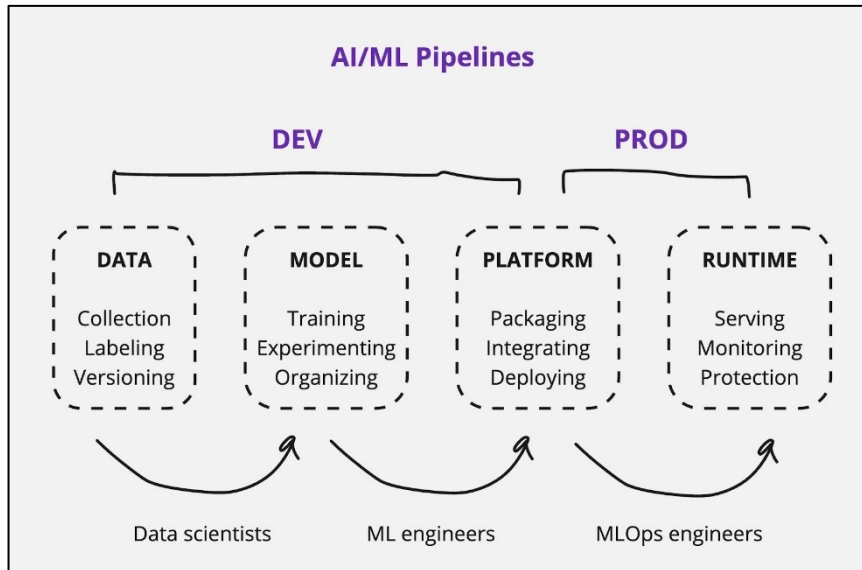
4.3. Security Challenges in AI/ML Pipelines

AI/ML pipelines are vulnerable to a range of security issues, including misconfigurations, vulnerabilities in container images, and potential attack scenarios.

Misconfigurations - Common issues include improper configurations in cloud services (e.g., S3 bucket permissions, insecure compute instances) and incorrect settings in containerized environments.

Vulnerabilities - These can be found in the software and underlying infrastructure used for model training and deployment, potentially leading to security breaches.

Attack Scenarios - Examples include adversarial attacks that manipulate AI model outputs or data poisoning attacks that corrupt the training data.

Financial Losses	Remediation costs, legal fees, fines.
IP Theft	Stolen AI models and algorithms.
Reputation Damage	Loss of customer trust.
Operational Disruptions	Downtime, productivity loss.
Legal Challenges	Non-compliance penalties.

4.4. Tools and Solutions for AI/ML Security

Several security tools are designed to address the unique challenges of AI/ML security. The issue lies in the fact that you need a different set of tools to achieve:

1. Visibility into AI/ML pipelines, identifying misconfigurations and vulnerabilities.

2. Protect AI models from various threats, including data breaches and adversarial attacks.

3. Secure machine learning models against adversarial threats and ensuring model integrity.

> While these tools offer valuable features, AccuKnox's ModelKnox platform excels with its integrated approach that combines visibility, risk management, and compliance tracking.
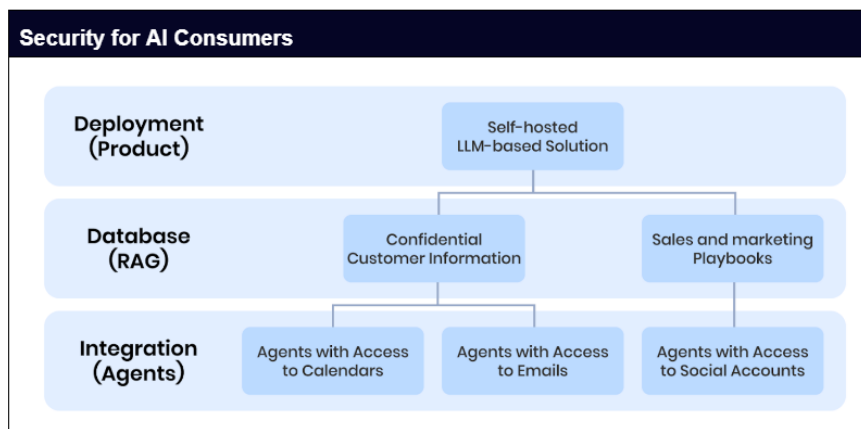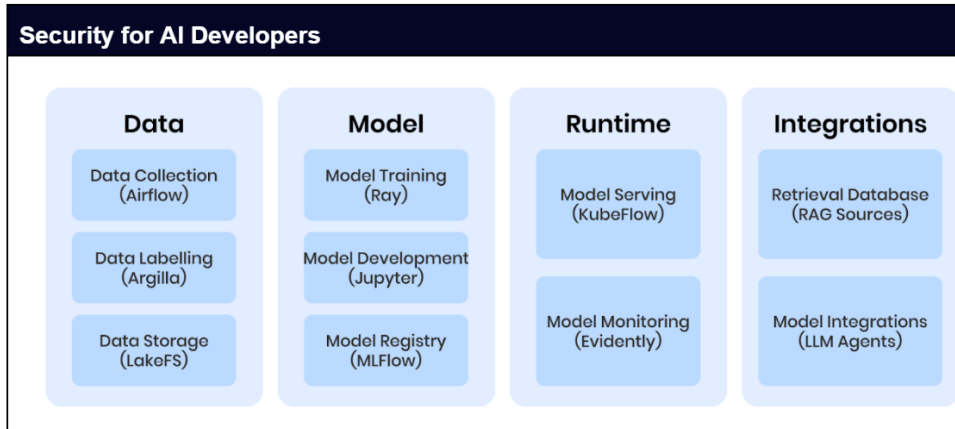
4.5. ModelKnox: Securing the AI/ML Lifecycle

In response to the growing cybersecurity threats surrounding AI and LLMs, AccuKnox is proud to announce the upcoming launch of ModelKnox, a cutting-edge solution designed to secure AI and LLM applications.

In the context of AI security, ModelKnox offers comprehensive protection against a range of threats:

AccuKnox ModelKnox - Zero Trust LLM Security

🔒 Cryptographic Security

Safeguard sensitive AI assets. **ModelKnox delivers the most advanced encryption of data in transit and data at rest for your sensitive AI intellectual property against unauthorized access.**

⚠️ Abusive Abuse

AI systems can be used to generate spam or conduct other fraudulent activities, among other things. **Abusive activity is spotted by ModelKnox through the analysis of behaviors and usage.**

📄 Compliance Mapping

Today, the market is already booming with complex regulations about AI. **ModelKnox automatically ensures compliance with leading frameworks, such as GDPR and CCPA, while keeping your operations in a legally safe zone.**

🔬 ML Security, Jupyter Notebook, and Other Workloads

Data scientists have for long had to use hostile environments like Jupyter notebooks to work, hence easily exposing security risks. **ModelKnox secures all the dev environments for machine learning, keeping sensitive data and code safe and isolated.**

🔒 Jailbreaking by Design

The whole idea is to deceive an AI into ignoring some of the programmed ethical restrictions—in other words, to force a model into doing what it isn't meant to do. It becomes very easy and simple when an end-user wants to fool a language model into generating malicious content. **ModelKnox ensures this won't occur through proper input validation and monitoring the model's output for anomalies.**

💻 SQL Injection

The main process somewhat involves attackers injecting inputted code from the users with the aim of governing the model's behavior. A very good example is the SQL command injection into text fields. **ModelKnox has put in place state-of-the-art sanitization on input and context-aware filtering, hence neutralizing threats of this nature.**

🔍 Pickle Scanning

In this case, the pickle files carry a path for malicious codes that are used for serialization of Python objects. **ModelKnox automatically scans all pickle files in use by AI pipelines and ensures they do not have any security vulnerabilities for your system.**

💥 Malware Scanning

AI model sometimes carries malware. For this reason, **ModelKnox involves special malware detection algorithms to scan the models before they are deployed in order to avert the proliferation of malicious code.**

ModelKnox aims to be a holistic framework to deliver observability, risk assessment, prioritization and mitigation for AI/LLM Models to address current and emerging threats.:

4.6. AI Workload Security Problems
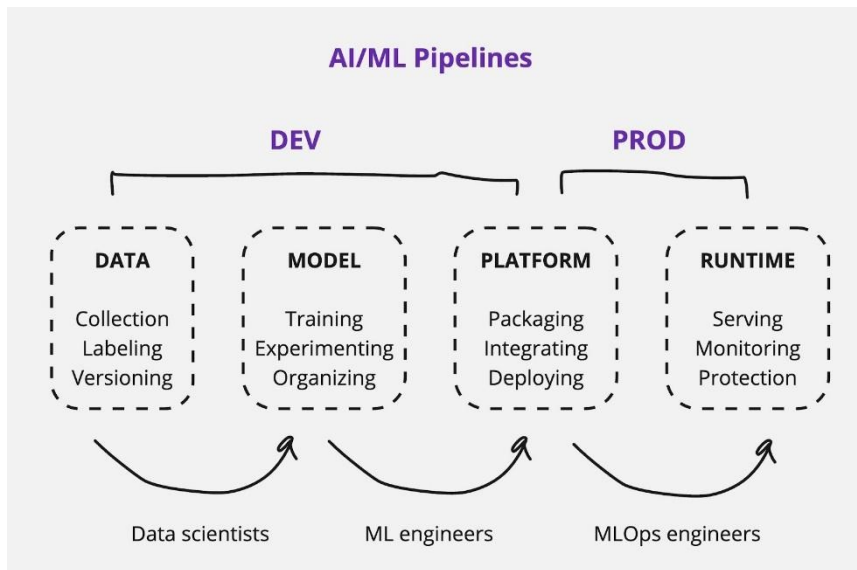
Modern AI workloads face a multitude of security challenges across different stages of the AI lifecycle, from data collection and model development to deployment and monitoring. These challenges include:

AI Workload Security Problem	Description
Insecure Model Registry (MLFlow)	Inadequate security measures in model registries like MLFlow can lead to model stealing, where unauthorized parties gain access to trained models.
Insecure Data Pipelines (Airflow)	Data pipelines orchestrated by tools like Airflow may be vulnerable to data leakage or poisoning attacks, compromising the integrity of the training data.
Insecure Orchestration (KubeFlow)	Orchestration platforms like KubeFlow may lack proper security controls, exposing the AI workload to various threats during deployment and runtime.
Insecure Training Pipelines (Ray)	Training pipelines managed by frameworks like Ray can be susceptible to data leakage or model stealing if not properly secured.

AccuKnox ModelKnox - Zero Trust LLM Security

AI Workload Security Problem	Description
Insecure Containers (Docker)	Insecure containers used for deploying AI models can lead to container exploitation, model hijacking, or other runtime attacks.
Insecure Development Environment (Jupyter)	Vulnerabilities in development environments like Jupyter Notebooks can enable model backdooring or remote-control attacks.
Insecure Network Connections	Unencrypted or insecure network connections can expose sensitive data and models to eavesdropping or man-in-the-middle attacks.
Insecure Supply Chain	Vulnerabilities or malicious components in the AI supply chain, including data sources, pre-trained models, and containers, can introduce risks.
Lack of Security Observability	Insufficient monitoring and observability mechanisms can hinder the detection and response to security incidents affecting AI workloads.
Insecure Models (DeepML or GenAI)	Inherent vulnerabilities in deep learning or generative AI models can lead to various attacks, such as model inversion, membership inference, or adversarial examples.

ModelKnox aims to be a holistic framework to deliver observability, risk assessment, prioritization and mitigation for AI/LLM Models to address current and emerging threats.

AccuKnox ModelKnox - Zero Trust LLM Security

AI Workload Security Solutions

Secure Inventory

Data Access Controls

Service Sandboxing

Training Access Controls

Container Image Scanning

DATA

DEPLOY

MODEL

RELEASE

RUNTIME

Development Environment Hardening

Network Segmentation

EVALUATE

MONITOR

Supply Chain Security

Security Observability

Model Security Testing

AI Developers and AI Consumers

Secure Infrastructure

Secure Models

Secure Data

Secure Deployment (PRODUCT)

Secure Database (RAG)

Secure Connections (AGENTS)

4.7. How ModelKnox Solves AI Workload Security Problems

4.7.1. Data Tasks and Data Risks

In the context of AI workloads, data tasks play a crucial role in ensuring the quality and integrity of the data used for training models. However, these tasks can also introduce various security risks if not properly secured.

12K Misconfigured Elasticsearch Buckets Ravaged by Extortionists

The cloud instances were left open to the public Internet with no authentication, allowing attackers to wipe the data.

38TB of data accidentally exposed by Microsoft AI researchers

Data Task	Security Risk	Tools Involved
Data collection	Data poisoning	Airflow
Data curation	Data leakage, data poisoning	Argilla, CleanLab, Labelbox, Label Studio
Data management	Data leakage, data poisoning	DVC, LakeFS

During the data collection phase, malicious actors can attempt to poison the data by injecting corrupted or malicious samples. This can lead to biased or compromised models. Airflow, a popular data pipeline orchestration tool, can be vulnerable to such attacks if not properly secured. In the data curation stage, where data is cleaned, labeled, and prepared for training, data leakage or poisoning attacks can occur. Tools like Argilla, CleanLab, Labelbox, and Label Studio, which assist in data curation, may be susceptible to these risks if their security measures are inadequate. Data management tasks, such as versioning, storage, and retrieval of data, can also be targeted by attackers. If the data management tools, like DVC (Data Version Control) or LakeFS (Data Lake Management), are not properly secured, data leakage or poisoning can occur, compromising the integrity of the AI workload.

4.7.2. Model Tasks and Model Risks

The development and management of AI models also involve various tasks and associated security risks.

Model Task	Security Risk	Tools Involved
Model training	Data leakage, model stealing	Ray, Metaflow, KubeFlow
Model development	Model backdooring, remote control	Jupyter, MLFlow
Model registry	Model stealing	ModelDB
Model packaging	Model stealing, model hijacking	MLFlow

During model training, data leakage or model stealing attacks can occur, where sensitive information or the trained model itself is accessed by unauthorized parties. Tools like Ray, Metaflow, and KubeFlow, which are used for distributed training and

orchestration, need to be secured to mitigate these risks. In the model development phase, vulnerabilities in the development environments, such as Jupyter Notebooks or MLFlow, can enable attacks like model backdooring or remote control. These attacks can allow adversaries to manipulate the model's behavior or gain unauthorized access to the system. Model registries, like ModelDB, store trained models and their metadata. If these registries are not properly secured, attackers can potentially steal models, leading to intellectual property theft or other malicious activities. Model packaging, which involves preparing the trained model for deployment, can also be a target for attacks like model stealing or model hijacking. Tools like MLFlow, used for model packaging and deployment, need to be hardened against such threats.

4.7.3. Platform Tasks and Platform Risks

The integration and deployment of AI workloads on platforms also introduce security risks that need to be addressed.

Platform Task	Security Risk	Tools Involved
Integrations	Remote control via trusted connections	LLM agents
Knowledge	Data exfiltration via RAG requests	RAG database
Model serving	Model hijacking, stealing, container exploitation	KubeFlow, Merlin, Tensorflow Serving, TorchServe
Observability	Attack hiding	Uptrain, Evidently

Integrations, such as LLM (Large Language Model) agents, can potentially be exploited to gain remote control over the system if the trusted connections are not properly secured.

Knowledge bases, like those used in Retrieval-Augmented Generation (RAG) systems, can be targeted for data exfiltration attacks if the RAG requests are not properly monitored and secured. During model serving, where trained models are deployed and exposed for inference, attacks like model hijacking, stealing, or container exploitation can occur. Platforms like KubeFlow, Merlin, Tensorflow Serving, and TorchServe need to be hardened against these threats. Observability tools, such as Uptrain and Evidently, which are used for monitoring and analyzing AI workloads, can potentially be exploited to hide or obfuscate attacks if their security measures are inadequate. To mitigate these risks, ModelKnox offers various data protection features, including container scanning, network segmentation, connection monitoring, access control, sandboxing, monitoring, and exposure control. By implementing these security measures, organizations can

safeguard their AI workloads and ensure the integrity and trustworthiness of their AI systems.

ModelKnox addresses these security challenges by providing a comprehensive solution for securing AI workloads throughout their lifecycle.

Secure Environment

ModelKnox enables secure management of AI workloads across development, training, and production environments, ensuring:

- Asset inventory and vulnerability management
- Supply chain security
- Secure communication channels

Secure Communication

ModelKnox protects connections for integrations, plugins, agents, and RAG (Retrieval-Augmented Generation) databases through:

- Sandbox isolation
- Network segmentation
- Anomaly detection

Secure Runtime

ModelKnox ensures the security of AI workloads in production by:

- Performing container scanning
- Enforcing security policies
- Leveraging threat intelligence

By addressing these critical security aspects, ModelKnox empowers organizations to develop, deploy, and operate AI workloads with confidence, mitigating risks and ensuring the integrity and trustworthiness of their AI systems.

4.8. Achieving Comprehensive Security with NVIDIA

4.8.1. AccuKnox Security Solution

AccuKnox security solution covers virtual machines (VMs), containers, and Kubernetes environments. Its capabilities include:

1. Risk Assessment: Evaluate the security posture of your infrastructure.
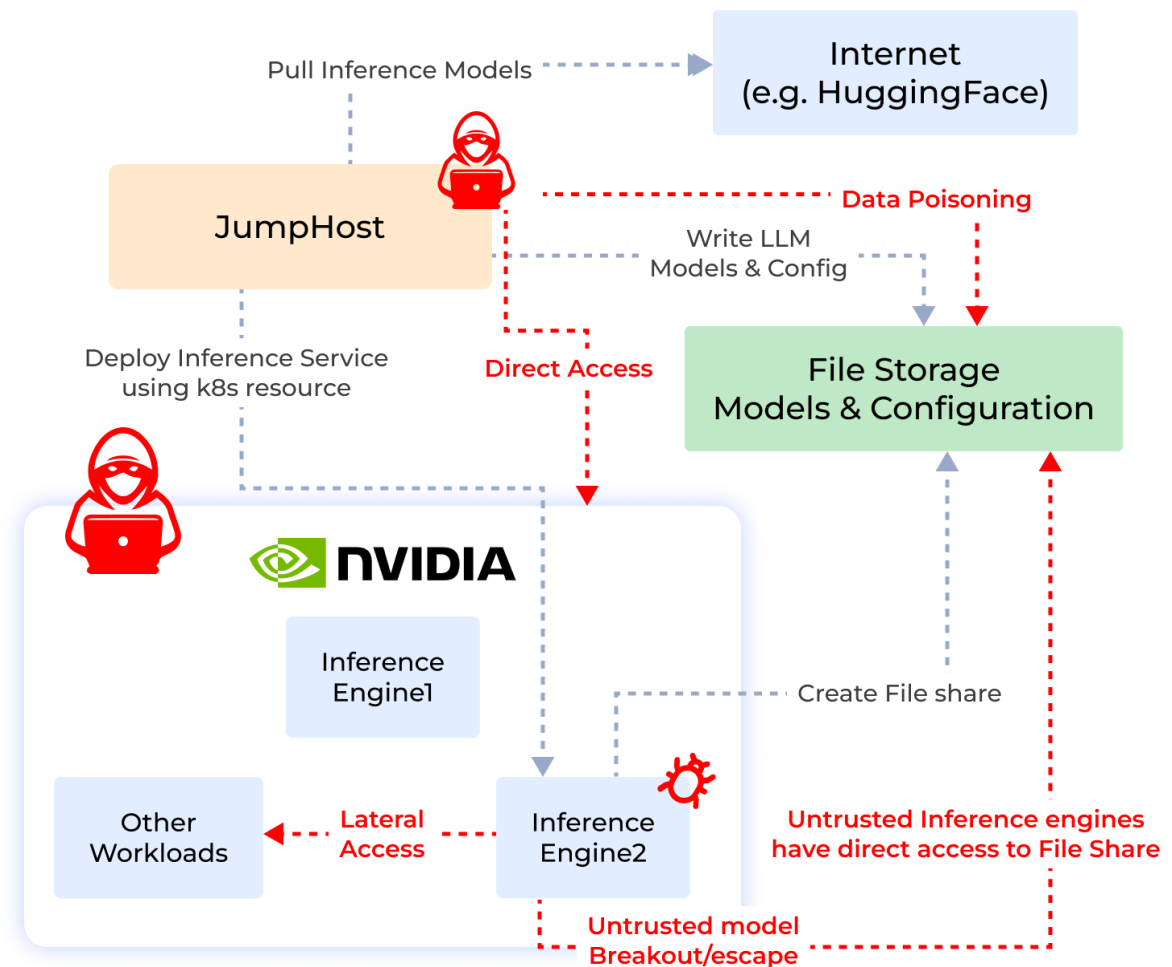
2. Security Hardening with KubeArmor: Enhance the security of your Kubernetes clusters.
3. Kubernetes Identity and Entitlements Management (KIEM): Manage identities and access controls within Kubernetes.
4. Application Behavior Analysis: Monitor and analyze the behavior of applications for anomalies.
5. Automated Zero Trust: Implement Zero Trust principles across your infrastructure.
6. Network Microsegmentation: Secure communication between workloads and resources.
7. Forensics and Auditing: Investigate and audit security incidents.
8. Compliance: Ensure adherence to industry standards like NIST, CIS, and STIGs.

AccuKnox supports both on-premises and managed deployments, catering to diverse organizational needs.

AccuKnox ModelKnox - Zero Trust LLM Security

4.8.1.1. Risks Associated with AI Deployments

1. Cryptomining Attacks: Malicious actors may target Kubeflow and TensorFlow deployments for crypto mining purposes.
2. Access Control for AI Data Lakes: Controlling access to sensitive data used for training AI models is crucial.

4.8.1.2. Limitations of Traditional Security Approaches

1. ML-based Anomaly Detection: These techniques may not be effective when applied to AI-based stacks due to the inherent complexity and variability of AI workloads.
2. Signature-based Detection: Predetermined signatures cannot effectively detect novel or evolving threats in AI environments.

ModelKnox also has security solution tailored for AI and LLM deployments:

1. Cryptomining Protection: AccuKnox not only detects but also prevents crypto mining attacks.
2. Sandboxing: Isolate and secure Jupyter notebooks, PyTorch apps, and other AI components.

```
apiVersion: security.kubearmor.com/v1
kind: KubeArmorPolicy
metadata:
  name: protect-jupyter
  namespace: jupyter
spec:
  selector:
    matchLabels:
      app: jupyterhub
      component: singleuser-server
  network:
    matchProtocols:
    - fromSource:
      - path: /usr/local/bin/python3.11
      protocol: udp
    - fromSource:
      - path: /usr/local/bin/python3.11
      protocol: tcp
  file:
    matchDirectories:
    - dir: /
      recursive: true
    - dir: /usr/local/bin/
      readOnly: true
    - dir: /bin/
      readOnly: true
    - dir: /usr/bin/
      readOnly: true
  process:
    matchDirectories:
    - dir: /usr/local/bin/
    - dir: /usr/bin/
    - dir: /bin/
  action: Allow
```

Only allow Python to use Network primitives

Only read access allowed to /usr/local/bin/ and /bin/folders

Execution allowed only from /usr/local/bin and /bin/folders, execution from every other path is denied

With Pre-emptive Mitigation

3. Zero Trust Policies: Implement robust Zero Trust policies to defend against future attacks.

4. Performance Optimization: AccuKnox's solution has minimal impact on the performance of AI stacks.

4.8.2. Specific Risks associated with ML stacks and How AccuKnox Steps In

1. PyTorch, TensorFlow

   a. Always execute untrusted models inside a sandbox (TensorFlow/security-policy)

2. Jupyter Notebooks

   a. Essentially allows remote command exec

   b. Sandboxing is a necessity.

3. Protecting data, models, checkpoints

   a. Audit access

   b. Limit access

AccuKnox ModelKnox - Zero Trust LLM Security

4. Platform risks
   a. K8s, VMs, Containerized deployments

AccuKnox provides comprehensive protection against these risks through:

1. Sandboxing with Preemptive Mitigation to isolate and secure AI components.
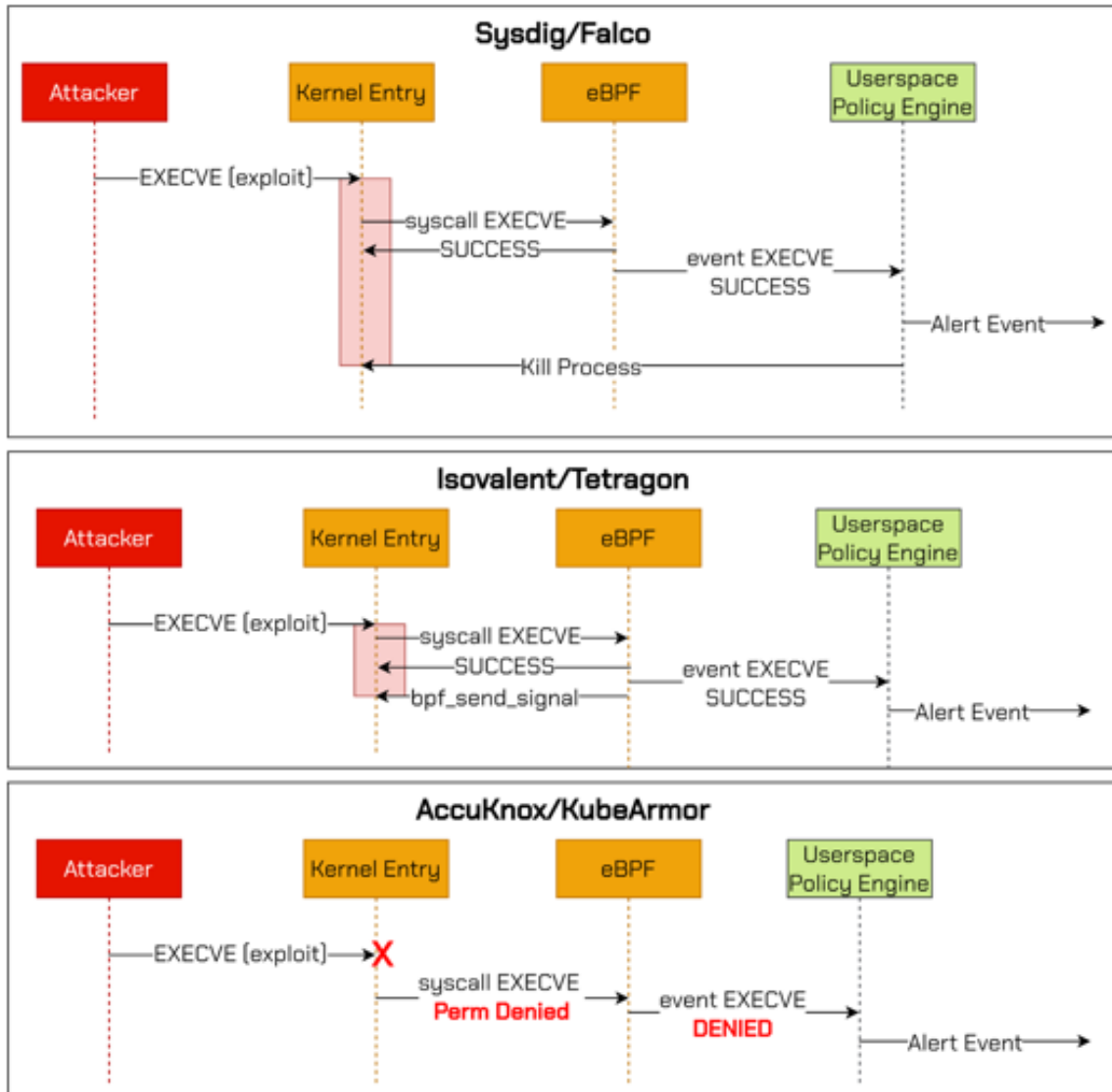
2. Software Bill of Materials (SBOM) Verification which validates the integrity of AI components.

3. Container Scanning for scanning containers for vulnerabilities and misconfigurations.

4. Malware Protection to detect and prevent malware infections.

5. Workload Hardening to shield AI workloads against potential vulnerabilities.

6. Data Fencing for controlling and restricting access to sensitive data used by AI workloads.

7. Network Micro Segmentation for securing communication between AI components and resources.

8. Sandboxing Untrusted Model Execution: ModelKnox offers a strong sandboxing mechanism to isolate and execute untrusted AI/ML models within a controlled environment. This ensures no possible threat leaves the sandbox to affect the overall system.
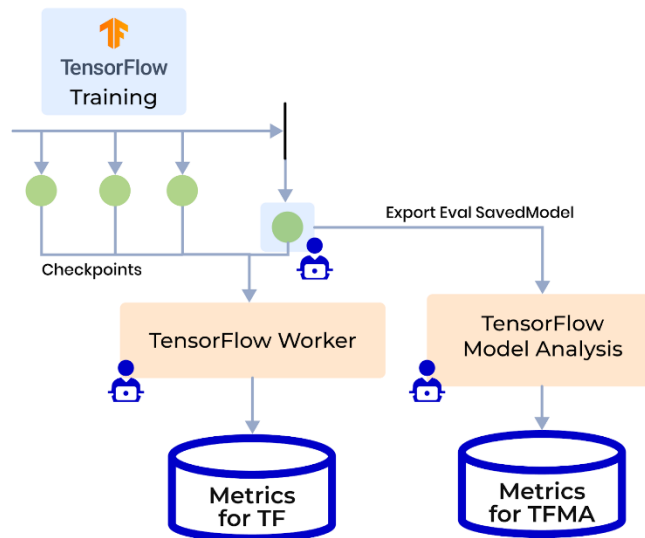
Even if the untrusted party only supplies the serialized computation graph (in form of a GraphDef, SavedModel, or equivalent on-disk format), the set of computation primitives available to TensorFlow is powerful enough that you should assume that the TensorFlow process effectively executes arbitrary code. One common solution is to allow only a few safe Ops. While this is possible in theory, we still recommend you sandbox the execution.

9. Data Fencing: ModelKnox implements data-fencing to ensure data-supply-poisoning attacks are protected. By restraining access to sensitive data and monitoring the data flow, ModelKnox maintains the quality of training datasets and model outputs.

10. Container Security: ModelKnox combines cutting-edge container security to prevent container breakouts and further unauthorized access into the host system. This is facilitated through the implementation of containerized deployment of inference engines with required access control.

AccuKnox ModelKnox - Zero Trust LLM Security

11. **Controlled Access to NKP (NVIDIA Kernel Provider):** Controlling the access to NKP from only the whitelisted jump hosts avoids exposure of GPU resources to unauthorized users or processes for AI/ML workloads.

12. **On-Premises/Hosted GPT-in-a-Box:** ModelKnox offers support for both on-premises and hosted deployment of GPT-in-a-Box, a secure, scalable way of running Large Language Model workloads, such as GPT-3.
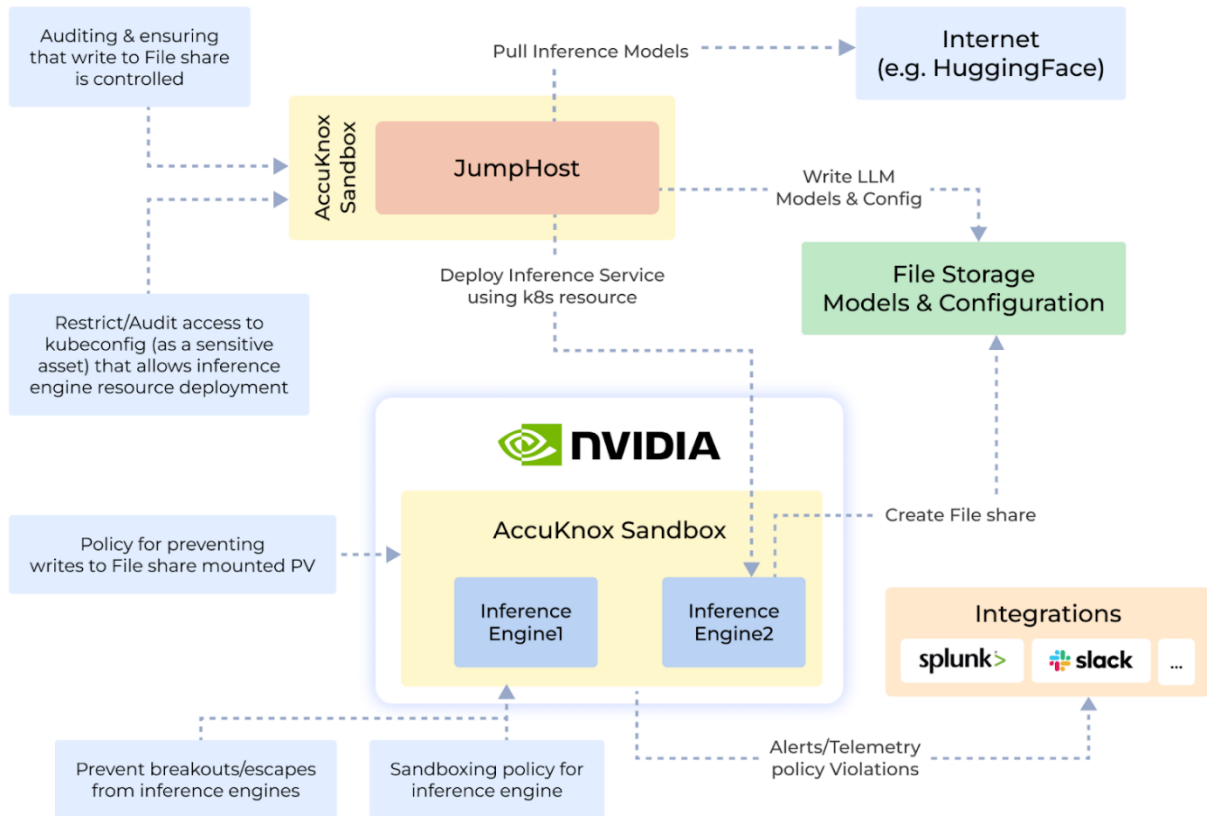
13. **Inference Engines and Models Admission Control:** ModelKnox ensures that only trusted inference engines or models are admitted into the production environment. If the trustworthiness or the provenance concerning the components is unknown, the untrusted components are bound to be deployed in a sandboxed or restricted run-time environment for further analysis and auditing.
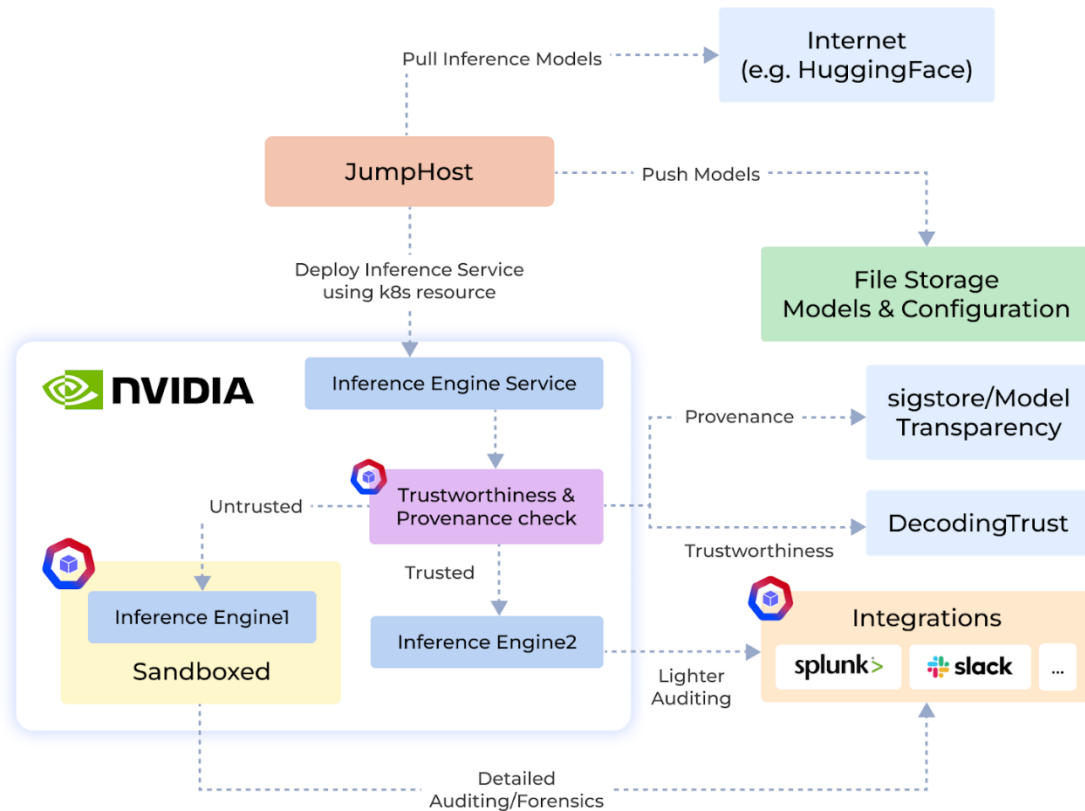
14. Extensive Auditing and Forensics Functionality: ModelKnox has rich auditing and forensics capabilities that enable organizations to monitor and investigate any suspicious activities or security incidents that might arise related to their AI/ML workloads.

15. Preemptive mitigation: It takes a larger and preemptive approach to mitigation rather than a traditional detect-and-respond approach. It identifies and enforces least-permissive policies for each application, thereby preventing potential security threats ahead of time.

16. Scalable and Performant Platform: ModelKnox has been designed as a full Kubernetes-native management console. It can scale resources both horizontally and vertically. It supports onboarding thousands of clusters using a single console, thereby providing seamless integration with on-premises and hosted Kubernetes deployments.

4.8.2.1. Admission Control of Inference Engines

AccuKnox is working on a feature that will enable admission control for inference engines and models before deployment

1. Validating Trustworthiness and Provenance: Verify the trustworthiness and provenance of inference engines and models before admitting them into the production environment.

2. Sandboxing Untrusted Components: If an inference engine or model is deemed untrustworthy, it can be deployed in a secure sandbox environment for further analysis and monitoring.

3. Detailed Auditing and Forensics: Enable detailed auditing and forensics for untrusted components to monitor their behavior and investigate potential threats.

4. Detailed Reporting: Provide detailed reports on the behavior of untrusted components.

5. Deployment Restrictions: AccuKnox may prevent the deployment of untrusted components altogether if the risk is deemed too high.
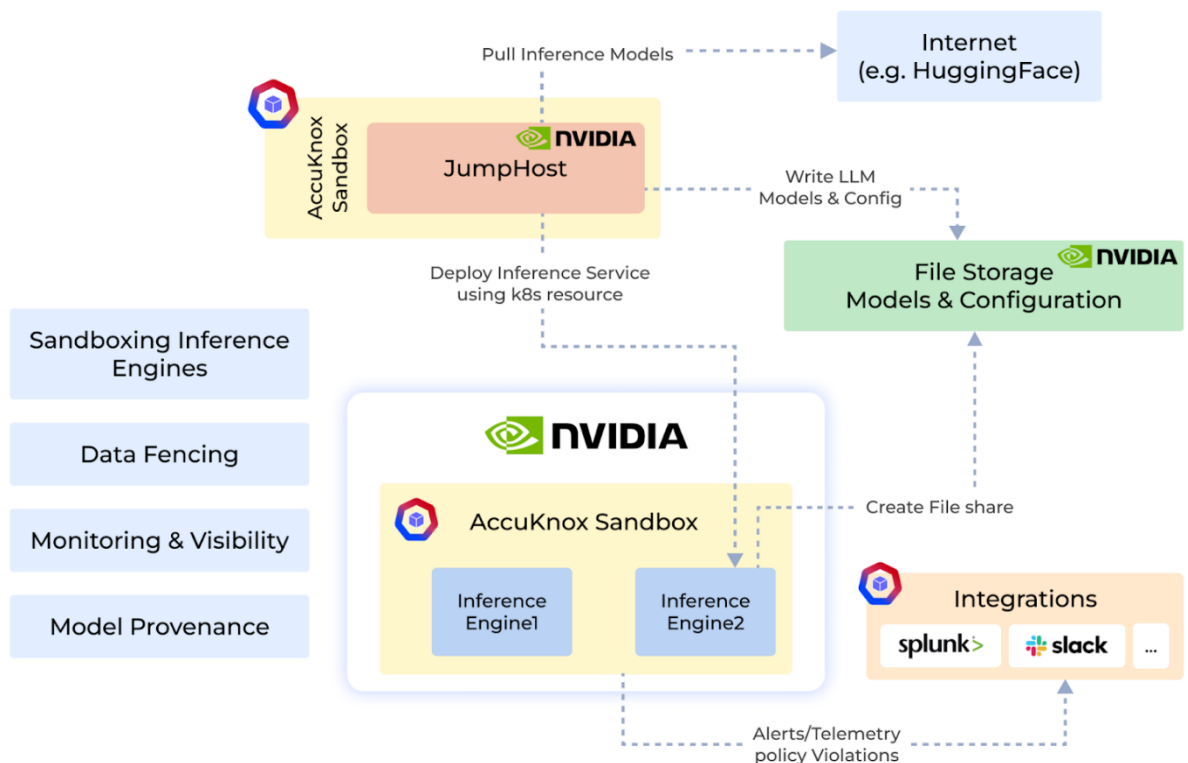
4.9. Securing CUDA Toolkit With AccuKnox

The CUDA Toolkit, a powerful software suite provided by NVIDIA, enables developers to leverage the computational power of GPUs for accelerating various applications. The risks associated with CUDA's usage have become increasingly significant with its widespread adoption. ModelKnox offers a comprehensive approach to mitigating these

risks and ensuring the secure deployment of CUDA-enabled applications with its cutting-edge security solutions.

4.9.1. Risks Associated with CUDA

Attackers who gain unauthorized access to a target platform can potentially inject malicious binaries and leverage the CUDA Toolkit to harness the computing power of GPUs for nefarious purposes. Additionally, the CUDA Toolkit itself may contain vulnerabilities that could be exploited by malicious actors. Furthermore, cryptocurrency miners often abuse the CUDA Toolkit's capabilities to mine digital currencies surreptitiously, consuming valuable computing resources.

4.9.2. Aspects of CUDA Security

To address these risks, ModelKnox focuses on four key aspects of CUDA security:

1. Access Control - Robust access controls to regulate and monitor the usage of the CUDA Toolkit.

2. Vulnerability Scanning - Regularly scanning the CUDA Toolkit for known vulnerabilities and applying necessary patches or mitigations.

3. Visibility and Monitoring - Comprehensive visibility into applications utilizing the CUDA Toolkit, monitoring their usage patterns, and detecting anomalous access attempts.

4. Sandboxing - Isolating and sandboxing applications to prevent unauthorized access to the CUDA Toolkit and its associated resources.

4.9.3. Visibility and Monitoring of CUDA Toolkit Usage

ModelKnox's visibility and monitoring capabilities offer granular insights into the applications leveraging the CUDA Toolkit. This includes identifying the specific applications accessing the CUDA library, the frequency of usage, and the processes interacting with the CUDA Toolkit. Additionally, ModelKnox can detect and raise alerts for any anomalous or unauthorized access attempts, enabling prompt incident response and mitigation.
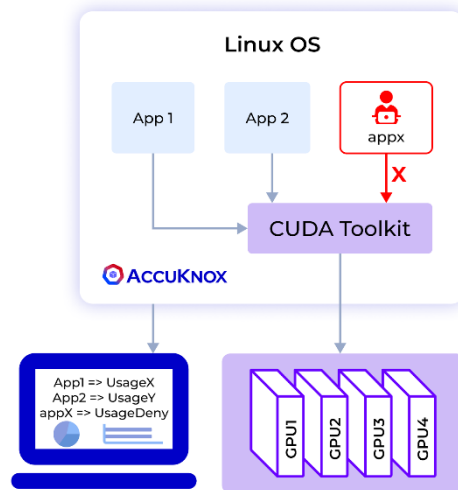
4.9.4. Sandboxing Access to CUDA Toolkit

The CUDA Toolkit is typically installed in system paths, such as /usr/local/cuda-12.4/lib64 and /usr/local/cuda-12.4/bin. ModelKnox's KubeArmor solution can sandbox the execution environment of applications, allowing only authorized applications to access the CUDA paths. This sandboxing approach leverages preemptive mitigation techniques, ensuring that applications operate within a least-permissive policy, thus

AccuKnox ModelKnox - Zero Trust LLM Security

minimizing the risk of unauthorized access or exploitation.
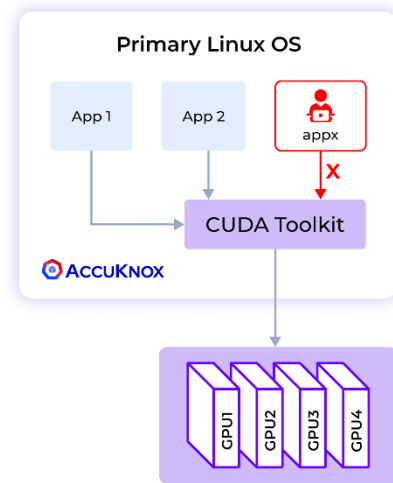
4.9.5. Vulnerability Scanning and Virtual Patching

ModelKnox's vulnerability scanning capabilities allow for the identification of known vulnerabilities within the CUDA Toolkit. Once identified, virtual patches can be applied to mitigate these vulnerabilities, providing a secure environment for CUDA-enabled applications without disrupting their functionality or performance.

Image Security Issues

Total Images	1,753

29	127	731	523	223	119

Total Malware: 689
Total Sensitive Data: 1064

Image Scanning Progress

Total Registered Images	4,657

- 3563 Scanned Images
- 67 Failed to Scan
- 1027 images in Queue
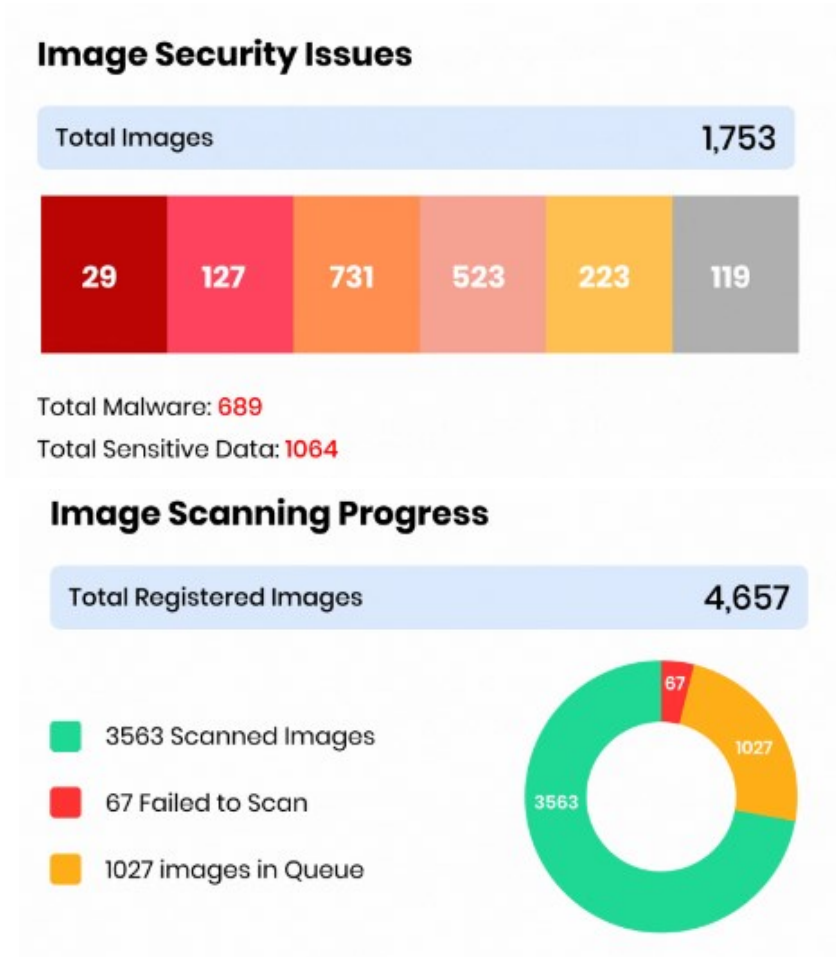
4.9.6. Deployment Simplicity and Versatility

ModelKnox's solutions are designed with deployment simplicity in mind, supporting both managed and unmanaged deployments across various platforms, including AWS, GCP, Azure, local Kubernetes clusters, and virtual machines. This versatility ensures a seamless integration into existing infrastructure, without impacting the runtime performance of GPU-accelerated applications.

Use ModelKnox's all-inclusive security solutions to deploy and run CUDA-enabled applications with confidence. This will reduce the risks of unauthorized access, exploits, and misuse of resources while preserving the high-performance computing capabilities offered by NVIDIA's CUDA Toolkit.

4.10. ModelKnox Feature Set and Walkthrough

ModelKnox delivers security at the Application and Infrastructure level during the Development and Deployment phases. The schematic and user experience is depicted in the following sections.
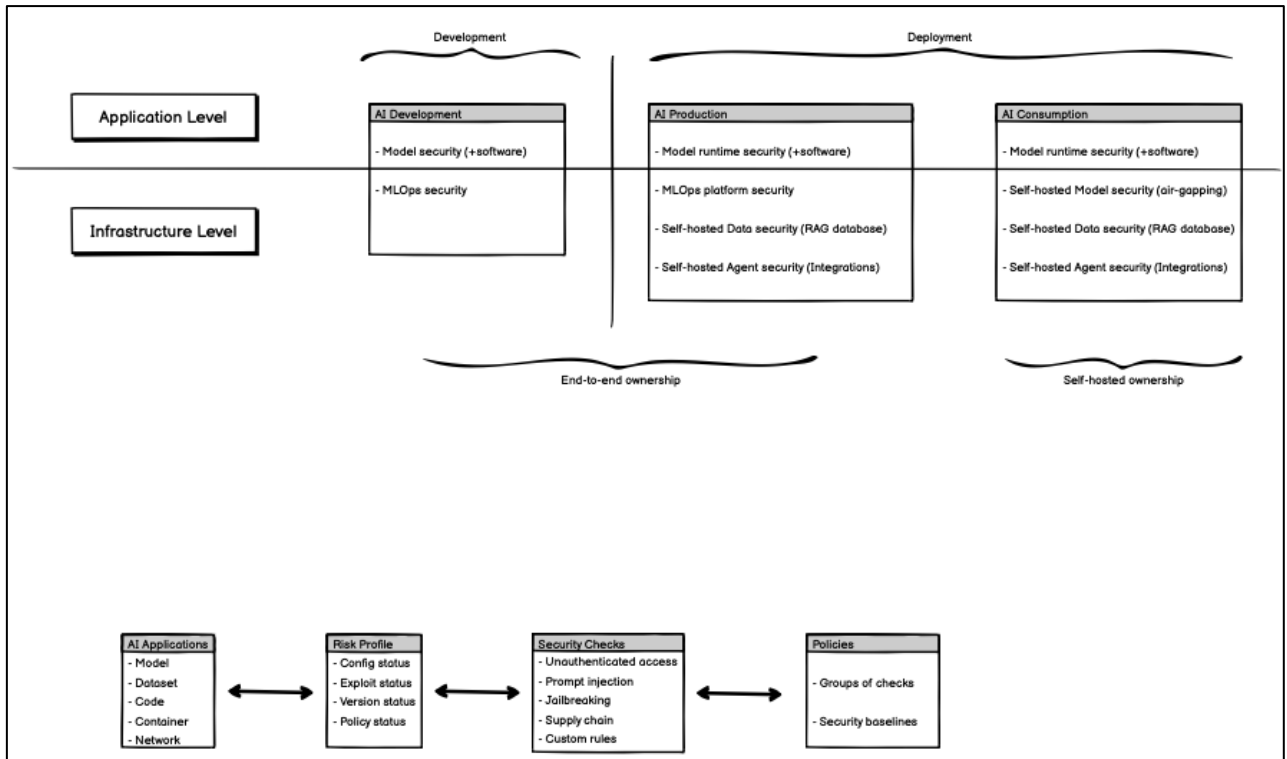
4.11. Compliance and Regulatory Requirements

Ensuring compliance with regulations such as the EU AI Act and NIST AI RMF is crucial for organizations operating in the AI/ML space. These frameworks provide guidelines for managing AI risks and ensuring the ethical use of AI technologies.

AccuKnox's AskAda Co-Pilot Assistant helps navigate compliance requirements and provides actionable insights to secure AI models. By aligning with these regulatory standards, ModelKnox enables organizations to stay ahead of evolving compliance demands and mitigate the associated risks.

4.12. ModelKnox Dashboard
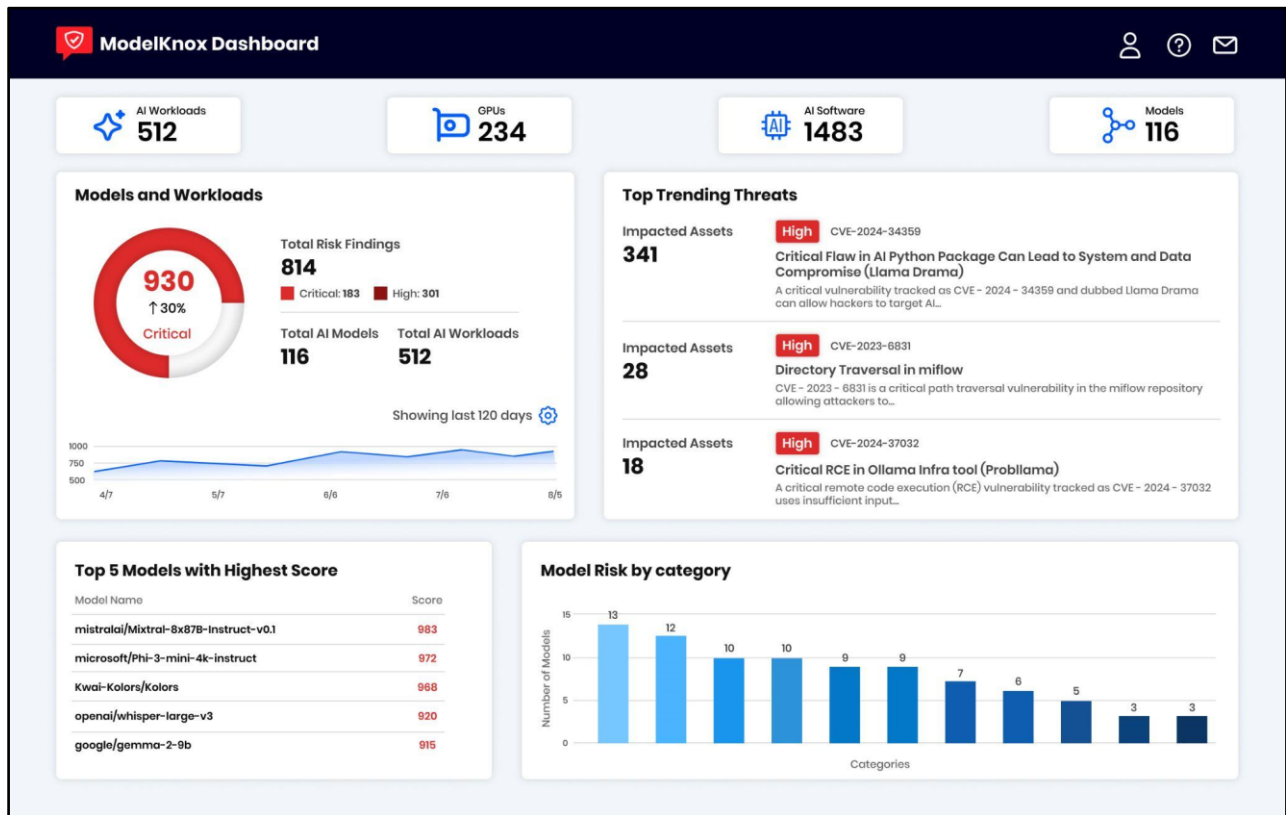
- Centralized Navigation: Top navigation for easy access to the main sections like Home, Models, and Vulnerabilities, all donning the most prominent platform branding.
- Overview of Critical Metrics: Tiles summarizing AI workloads, GPU usage, and status of models provide a quick snapshot of essential data.
- Risk Assessment Gauge: A center gauge providing overall security posture; included is a detailed breakdown of the associated risks by severity.
- Trend Analysis & Threat Monitoring: Sections providing historical risk trend tracking and identification of top security threats to keep users updated with newly emerging vulnerabilities.
- Top Risky Models: A table identifying AI models with the highest risk scores, enabling rapid identification of vulnerable assets.
- Categorization of Model Risk: This visualizes heterogeneous risk types like fuzzing and adversarial attacks, providing specific details about some threats.
- Compliance Tracking: A dedicated section for monitoring adherence to internal and industry policies, ensuring compliance, and identifying issues.
- User-Centric Features: Search functionality, profile management, and customizable settings for easy navigation and deeper analysis.

4.13. AI Asset Inventory Dashboard

This view addresses the problem of distributed resources in cloud environments. Get very clear metrics of models, datasets, pipelines, and jobs in understanding the scale of their AI operations. In-depth analysis of datasets will facilitate tracking and management, thus reducing the risks associated with using outdated or unauthorized data in AI procedures.

LIST VIEW

DETAILED VIEW

This provides visibility about multiple versions of the model. One can maintain the production and pre-production versions with associated issues and vulnerabilities. Crystal clear representation of container and network information helps the DevOps team to identify potential security risks right there and take remedial measures so that the model deployment process runs smoothly.

4.13.1. Model Issues Prioritization

This dashboard deals with the very important problem of AI security issue monitoring. Intuitive visualization through distribution and the severity of the issues are shown using a sankey diagram; action-insight is provided in the list in detail. From this view, security teams can focus on prioritizing vulnerabilities to efficiently improve the integrity of overall AI systems.
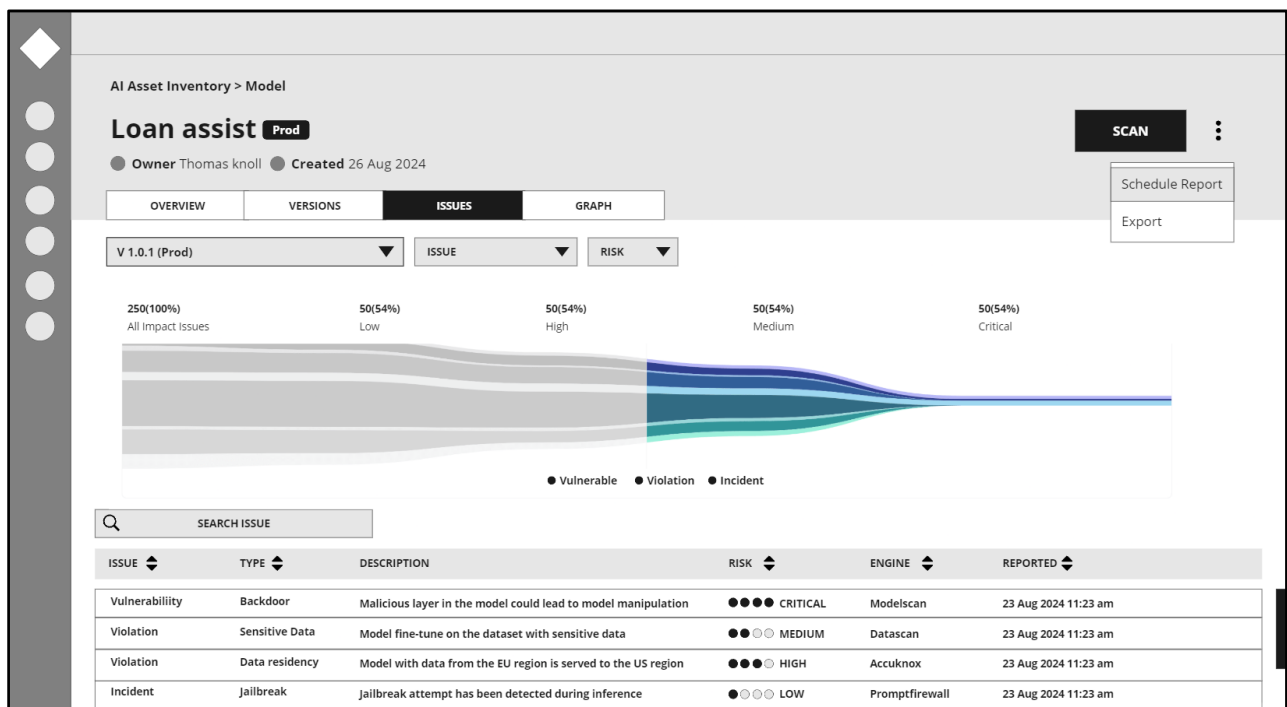
4.13.2. Pipeline Visualization

This graph view solves the problem of understanding complex AI pipelines. It helps in tracing the flowing data from source to deployment in a more visual way, with identification of bottlenecks and security vulnerabilities. It will also utilize color-coding to help identify production versus pre production environments for better resource and risk management.
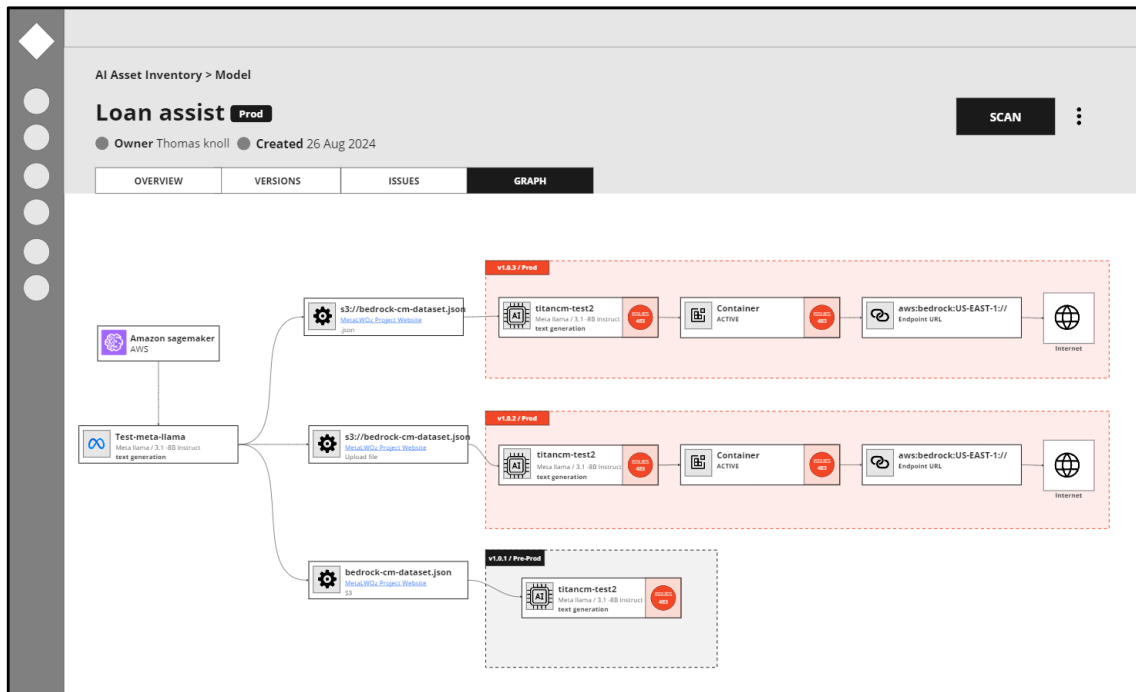
4.14. ModelKnox caters to varying user personas

1. Security Architects: Focus on risk management by providing granular visualization of vulnerabilities across models and pipelines.

2. MLOps & Cloud Engineers: Build visibility for multi-cloud AI/ML pipelines in workloads, misconfigurations, and infrastructure.

3. Data Scientists: Empower data science teams to quickly identify and address security issues in their AI/ML models and pipelines. ModelKnox provides detailed visibility into model performance, vulnerabilities, and compliance, helping data scientists ensure the integrity and security of their AI systems.

4. Compliance Officers: Enable effective governance and risk management by aligning AI/ML security practices with regulatory frameworks like the EU AI Act and NIST AI RMF. ModelKnox's compliance tracking capabilities and AskAda co-pilot assistant help compliance teams monitor policy adherence and remediate any issues.

By catering to these diverse user roles, ModelKnox ensures that the entire organization, from security and cloud teams to data scientists and compliance officers, can collaborate effectively to secure the AI and LLM landscape.

4.15. What makes ModelKnox unique?

1. Multi Cloud Visibility: Single view into AI/ML pipelines across GCP, Azure, and AWS via the Graph interface.

2. Runtime Protection: Apply Industry's leading runtime protection to mitigate against AI based zero-day attacks.

3. Centralized Risk Insights: Overall assessment of cloud misconfigurations, AI vulnerabilities, and container security.

4. AskAda Integration: Co-Pilot for AI security companion.

5. AI Model Performance Monitoring: Real-time monitoring of AI model performance, correlated with security metrics.

6. Model Knox Enhancements: Graph and list views for multi-cloud models, including detailed metadata such as Model ARN, base model, and customization types.

7. Advanced Analytics: Predictive analytics with advanced capabilities of threat detection.

8. Security Visualization: Graphical views of pipelines with detected vulnerabilities and misconfigurations.

9. Compliance tracking: Standards and regulations like the EU AI Act and NIST AI RMF are added into our compliance tracking roadmap.

AccuKnox ModelKnox - Zero Trust LLM Security
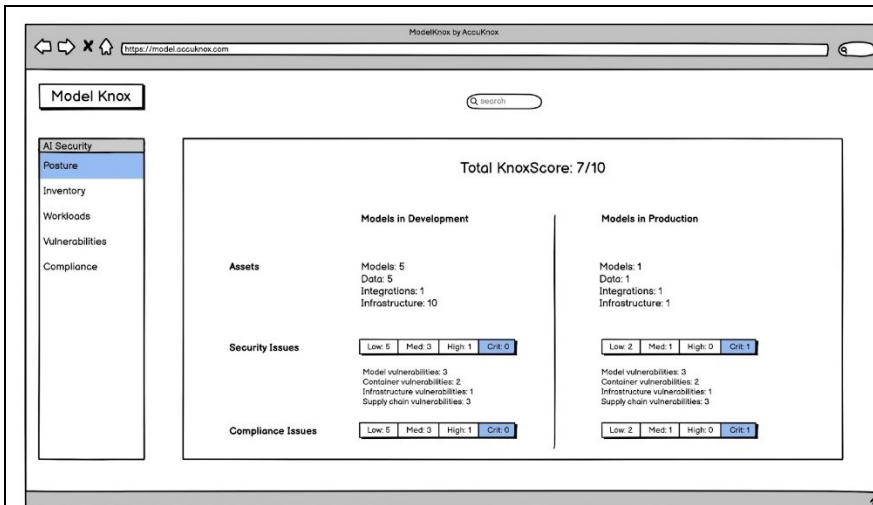
The ModelKnox dashboard offers a detailed overview of the organization's AI security posture, including inventory, workloads, vulnerabilities, and compliance. Users can view vulnerabilities found in models under development and in production, categorized into model, container, infrastructure, and supply chain vulnerabilities. The dashboard also provides specifics about each asset, including the type and source of vulnerability information, helping users to pinpoint areas that need immediate attention.
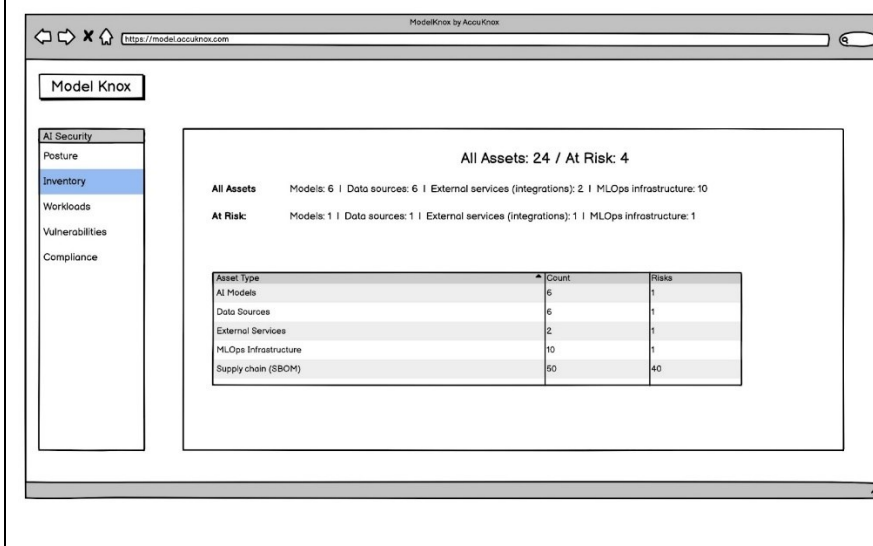
The Inventory section provides an overview of various assets within the AI/ML ecosystem, including AI models, data sources, external services, MLOps infrastructure, and the supply chain. Users can see the total number of assets and those considered "At Risk," enabling them to quickly identify and prioritize areas that may need remediation to enhance security.

AccuKnox ModelKnox - Zero Trust LLM Security
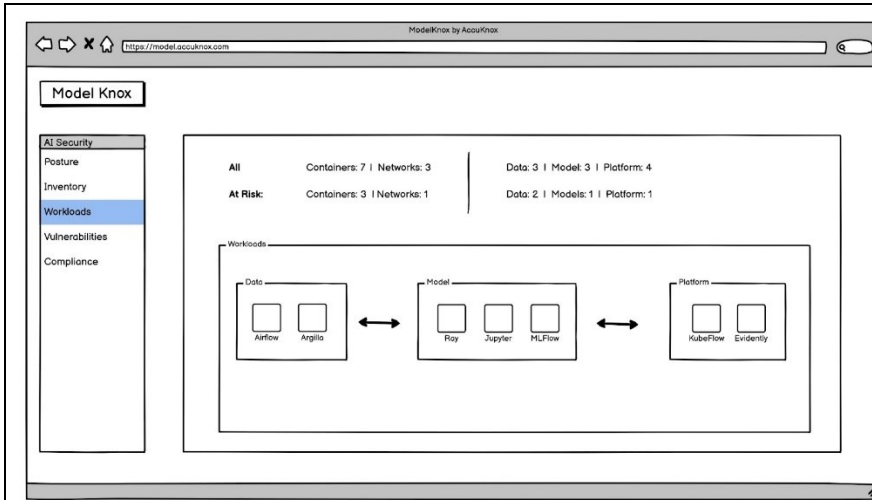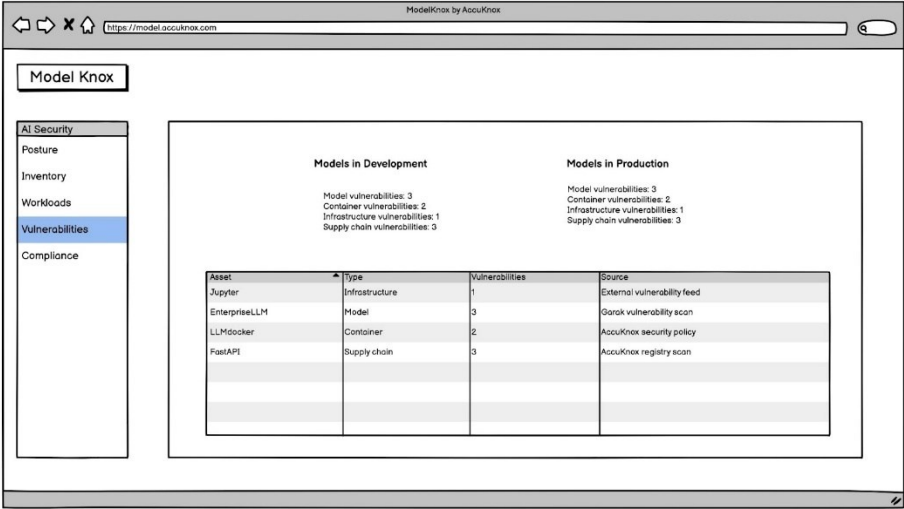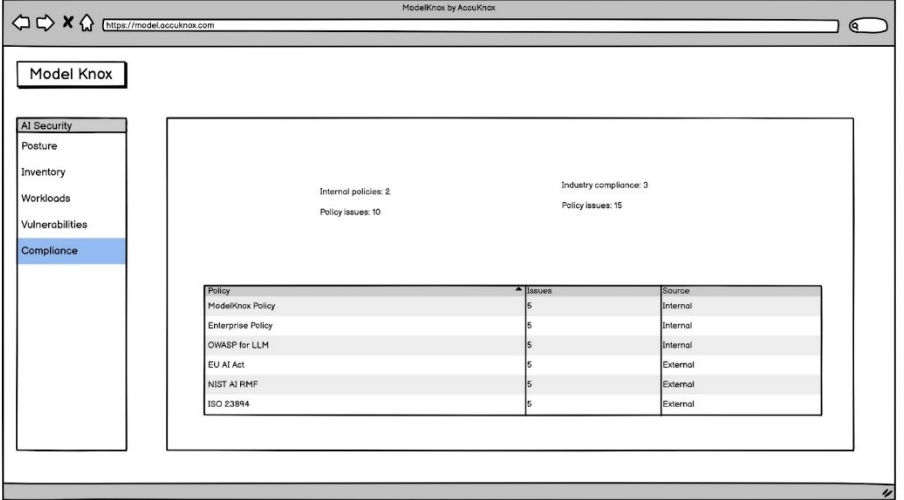
This section provides a visual representation of the relationships between different AI/ML workload components, categorized into Data, Model, and Platform groups. For instance, it shows how workloads like Airflow and Argilla (Data group) are connected, and how components such as Ray, Jupyter, and MLFlow (Model group) interact with each other. This visualization helps users understand the dependencies and interactions within their AI/ML infrastructure.

In the "Workloads" section, users can get a snapshot of AI/ML workloads and their associated risks. The dashboard lists the total number of containers, networks, data sources, models, and platforms, along with the number of assets considered "At Risk" in each category. It helps users identify workloads that may pose potential security risks, allowing for proactive investigation and remediation.

AccuKnox ModelKnox - Zero Trust LLM Security

Asset	Type	Vulnerabilities	Source
Jupyter	Infrastructure	1	External vulnerability feed
EnterpriseLLM	Model	3	Garak vulnerability scan
LLMdocker	Container	2	AccuKnox security policy
FastAPI	Supply chain	3	AccuKnox registry scan

Models in Development
Model vulnerabilities: 3
Container vulnerabilities: 2
Infrastructure vulnerabilities: 1
Supply chain vulnerabilities: 3

Models in Production
Model vulnerabilities: 3
Container vulnerabilities: 2
Infrastructure vulnerabilities: 1
Supply chain vulnerabilities: 3

The "Posture" section offers a comprehensive view of the overall AI security posture, quantified by the Total KnoxScore. It is divided into "Models in Development" and "Models in Production," listing the number of assets and security issues categorized by severity levels (Low, Medium, High, and Critical). This allows users to assess the security posture of their AI/ML infrastructure effectively, ensuring that both development and production environments are secure and compliant.

Internal policies: 2
Policy issues: 10

Industry compliance: 3
Policy issues: 15

Policy	Issues	Source
ModelKnox Policy	5	Internal
Enterprise Policy	5	Internal
OWASP for LLM	5	Internal
EU AI Act	5	External
NIST AI RMF	5	External
ISO 23894	5	External

The compliance section showcases metrics for internal policies, policy issues, industry compliance, and associated policy issues. A table below lists different policies, their issue counts, and indicates whether they are internal or external sources. This is where you can get a unified view for all policy types and keep up with compliances/policies/audits

With ModelKnox, you can easily manage and monitor your application components, including models, data, code, containers, and networks. Our dashboards, purposefully designed to be simple and intuitive, provide real-time visibility of potential security risks: prompt injection, model architecture vulnerability, and misconfigurations that expose data breaches or policy breaches.

We harness industry-best security engines and supplement them with custom policies to automate threat detection and remediation. ModelKnox proactively scans your applications, pinpoints issues, and gives actionable insights for mitigating risks effectively. Our solution helps you say miles ahead of emerging threats, while making

sure that the environment for your mission-critical applications is secure, compliant, and kept away from the network.

To give you a more tangible idea, from the All LLM Applications view, you get a single pane of glass that offers visibility into the kind of problems you are experiencing across your entire LLM portfolio, disaggregated by severity levels. Just with a glance, be aware of critical vulnerabilities, like insecure model hashes or exposures of sensitive data. But ModelKnox doesn't just bring issues to your attention – it provides you with the ability to drill down and actually do something about them. From the \"Workloads\" view, we present you with all the components of your LLM deployments: models and data, containers, and networking, giving you conformity about your asset landscape. Posture, our analysis, takes it one step further by displaying a granular-level breakdown of risks by severity level and target component. This high-level view allows you to easily see where attention needs to be directed, as with the multiple high-risk issues identified for the model and data components of the CustomFinanceLLM application.

AccuKnox ModelKnox - Zero Trust LLM Security

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Posture > CustomFinanceLLM

System

Owner >	John Connor	
Stage >	Dev	Prod
Risk >	Low	Medium	High	Critical
Platform >	Self-managed	Azure	AWS

Workload

Model >	meta-llama3-8B.gguf
Data >	s3://finance-customer-emails
Code >	tensorflow-sbom-requirements.txt
Container >	python-ubuntu-chainguard
Network >	prod-env-model-to-rag

Transparency

Model card	
Data sheet	
Risk appetite	

Todo:

Pull data about foundational models

Public benchmarks apis

Tabs: All | Model | Data | Code | Container | Network

Target	Issue	Type	Engine	Risk
Model	Vulnerable to jailbreaking	LLM security	Garak vuln scanner	High
Model	Mismatching model hash	Supply chain security	SbomScan	Critical
Data	Sensitive data found in RAG	RAG security	DataScan	High
Data	Potentially poisoned dataset	LLM security	DataScan	High
Code	Vulnerable Lanhchain version	Supply chain	SbomScan	Critical
Container	Outdated Ubuntu container	Supply chain	AccuKnox	Medium
Network	Unrestricted connections between containers	Environment	AccuKnox	Medium
Policy	Model and data in different regions	Policy violation	Environment	Low
Inference	Sensitive information disclosure	LLM Firewall	ContentScan	High

---

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Issues > Prompt Injection

Issue

Type >	LLM Security	
Class >	Manipulation	
Scenario >	Attacker can bypass built-in security and safety controls to manipulate LLM behaviors	
Advice >	Implement LLM guardrails or fine-tune safety and security capabilities of the model	

Info

Engine >	Garak scanner
Check >	probes / promptinject.py

Included in policies:

- Garak checks
- ModelKnox
- OWASP LLM Top 10

Todo:

Security analytics for individual checks

Found in Applications

Target	Issue	Risk
CustomFinanceLLM	Prompt injection	High
AzureChatGPTAPI	Prompt injection	High

AccuKnox ModelKnox - Zero Trust LLM Security

Our dashboard also hosts our OWASP LLM Top 10 policy, targeting critical security issues related to Large Language Models, including prompt injection, insecure output handling, and model theft. Up next is our Agent Security policy: The second image depicts our Agent Security policy, focusing on the runtime environment hardening of your AI agents. These critical areas cover the containerization, networking, authentication, and more for your AI applications to operate in a secure and resilient infrastructure. With ModelKnox, you are in control with more perspective into your AI environment, allowing you to stay a step ahead of risks before they spiral. The simple-to-use platform and customizable policies help you tune security measures to your requirements. Its user-centric interface allows you to understand what is happening across all applications at any given time.

AccuKnox ModelKnox - Zero Trust LLM Security

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Todo:
Link to individual checks

Policies > OWASP LLM Top 10

Issue

Type >   LLM Security

Class >   Industry Policy

Info

Engine >   Garak scanner

Checks in Policy

Requirement	Found in Applications
LLM01: Prompt Injection	CustomFinanceLLM AzureChatGPTAPI
LLM02: Insecure Output Handling	
LLM03: Training Data Poisoning	
LLM04: Model Denial of Service	
LLM05: Supply Chain Vulnerabilities	
LLM06: Sensitive Information Disclosure	
LLM07: Insecure Plugin Design	
LLM08: Excessive Agency	
LLM09: Overreliance	
LLM10: Model Theft	

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Todo:
Link to individual checks

Policies > ModelKnox Top 7

Issue

Type >   LLM Security

Class >   Custom Policy

Info

Engine >   ModelKnox scanner

Checks in Policy

Requirement	Found in Applications
KNX01: Insecure Model	CustomFinanceLLM AzureChatGPTAPI
KNX02: Insecure Dataset	
KNX03: Insecure Container	
KNX04: Insecure Networking	
KNX05: Insecure Data Retrieval	
KNX06: Insecure Library	
KNX07: Insecure Interaction	

AccuKnox ModelKnox - Zero Trust LLM Security

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Policies > Air-Gapped Environment

Issue

Type > LLM Security

Class > Custom Policy

Info

Engine > AccuKnox CNAPP

Todo:
Link to individual checks

Checks in Policy

Requirement	Found in Applications
AIR01: Hardened Container	
AIR02: Hardened Networking	
AIR03: Hardened Authentication	
AIR04: ...	
AIR05: ...	
AIR06: ...	
AIR07: ...	

---

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Policies > Agent Security

Issue

Type > LLM Security

Class > Custom Policy

Info

Engine > AccuKnox CNAPP

Todo:
Link to individual checks

Checks in Policy

Requirement	Found in Applications
AIR01: Hardened Container	
AIR02: Hardened Networking	
AIR03: Hardened Authentication	
AIR04: ...	
AIR05: ...	
AIR06: ...	
AIR07: ...	

AccuKnox ModelKnox - Zero Trust LLM Security

Screen 1

ModelKnox by AccuKnox

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Assets > Workloads

Todo:
Visualize like in AccuKnox

Application	Asset	Info
CustomFinanceLLM	Model	meta-llama3-8B.gguf
CustomFinanceLLM	Data	s3://finance-customer-emails
CustomFinanceLLM	Code	tensorflow-sbom-requirements.txt
CustomFinanceLLM	Container	python-ubuntu-chainguard
CustomFinanceLLM	Network	prod-env-model-to-rag
AzureChatGPTAPI	Model	openai-chatgpt-4
AzureChatGPTAPI	Data	customer-support-chats.json
AzureChatGPTAPI	Code	N/A
AzureChatGPTAPI	Container	azure-openai-managed
AzureChatGPTAPI	Network	N/A

Screen 2

ModelKnox by AccuKnox

https://model.accuknox.com

Model Knox

Posture
Inventory
Workloads
Vulnerabilities
Compliance

Assets > CustomFinanceLLM

Todo:
Map ML container ecosystem

Development

Data: Airflow, Argilla

Model: Ray, Jupyter, MLFlow

Platform: KubeFlow, Evidently

Production

Application	Asset	Info	Stage
CustomFinanceLLM	Model	meta-llama3-8B.gguf	Development
CustomFinanceLLM	Data	s3://finance-customer-emails	Development
CustomFinanceLLM	Code	tensorflow-sbom-requirements.txt	Development
CustomFinanceLLM	Container	python-ubuntu-chainguard	Development
CustomFinanceLLM	Network	prod-env-model-to-rag	Development

5. LLM AI Cybersecurity & Governance Checklist

Adversarial Risk

☐ Scrutinize how competitors are investing in artificial intelligence

☐ Investigate the impact of current controls

AccuKnox ModelKnox - Zero Trust LLM Security

- ☐ Update the Incident Response Plan and playbooks for GenAI-enhanced attacks and AIML-specific incidents

AI Asset Inventory

- ☐ Catalog existing AI services, tools, and owners
- ☐ Include AI components in the Software Bill of Material (SBOM)
- ☐ Catalog AI data sources and sensitivity
- ☐ Establish if pen testing or red teaming of deployed AI solutions is required
- ☐ Create an AI solution onboarding process
- ☐ Ensure skilled IT admin staff is available

AI Security and Privacy Training

- ☐ Engage with employees on LLM initiatives
- ☐ Establish culture of open communication on AI use
- ☐ Train users on ethics, responsibility, legal issues
- ☐ Update security awareness training to include GenAI threats
- ☐ Include training for DevOps and cybersecurity for any adopted GenAI solutions

Governance

- ☐ Establish AI RACI chart
- ☐ Assign AI risk, assessments and governance responsibility
- ☐ Establish data management policies
- ☐ Create an AI Policy
- ☐ Publish an acceptable use matrix for GenAI tools
- ☐ Document data sources and management from LLM models

Legal

- ☐ Confirm product warranties for AI
- ☐ Review and update terms and conditions for GenAI
- ☐ Review AI EULA agreements
- ☐ Modify end-user agreements

AccuKnox ModelKnox - Zero Trust LLM Security

- ☐ Review AI-assisted tools used for code
- ☐ Review risks to intellectual property
- ☐ Review contracts with indemnification provisions
- ☐ Review liability for AI systems
- ☐ Review insurance coverage
- ☐ Identify copyright issues
- ☐ Ensure agreements for contractors and AI use
- ☐ Restrict use of GenAI where IP or rights may be an issue
- ☐ Assess AI for employee management/hiring
- ☐ Ensure proper consent for sensitive data collection

Regulatory

- ☐ Determine AI compliance requirements
- ☐ Determine compliance for employee monitoring and decision systems
- ☐ Determine compliance for facial recognition and video analysis
- ☐ Review AI tools for hiring/employee management

Using or Implementing Large Language Model Solutions

- ☐ Threat model LLM components and architecture
- ☐ Verify data security
- ☐ Implement access controls
- ☐ Require rigorous control of training data, pipelines, models, algorithms
- ☐ Evaluate input validation and output filtering
- ☐ Map workflows, monitoring, response
- ☐ Include testing in production release
- ☐ Check for vulnerabilities in LLM model/supply chain
- ☐ Investigate impact of attacks on LLM
- ☐ Request audits for third-party providers

AccuKnox ModelKnox - Zero Trust LLM Security

☐ Update incident response playbooks

Testing, Evaluation, Verification, and Validation (TEVV)

☐ Establish continuous TEVV through AI lifecycle

☐ Provide executive metrics on AI model functionality, security, reliability, robustness

Model Cards and Risk Cards

☐ Review model's model card

☐ Review risk card if available

☐ Track and maintain model cards

5.1.1. AccuKnox Zero Trust Cloud-Native Application Protection Platform (CNAPP)

Developed in partnership with the prestigious R&D innovator, AccuKnox is the most comprehensive Zero Trust CNAPP::

1. Cloud Security Posture Management (CSPM): Scan cloud accounts, identify misconfigurations, and provide remediation guidance.

2. Application Security Posture Management (ASPM), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Infrastructure as Code (IaC) Scanning and Tool Integrations: Analyze and secure applications and infrastructure code.

3. Cloud Workload Protection Platform (CWPP): Scan workload images, assess risks in VMs, containers, and Kubernetes environments, and provide remediation recommendations.

4. Runtime Security: Differentiated offering for runtime security and threat detection.

5. Identity and Entitlements Management: Manage identities and access controls across your infrastructure.

6. Governance, Risk and Compliance (GRC): Compliance with 30+ Regulatory compliance frameworks like SOC2, NIST, CIS, MITRE, STIG, ISO, HIPAA, GDPR, etc.

With ModelKnox we are extending it to support AI/LLM Model Security.

6. Why AccuKnox?

5 Reasons

AccuKnox is Your Zero-Trust Cloud Security Tool For "Build to Runtime" Security

⊘ **Effortless** — Agentless: Onboard, Detect, and Protect in minutes!

⊘ **Extensive** — Secures (K8s) assets; Multi-cloud and On-prem

⊘ **Effective** — Inline remediation to prevent "Zero Day" attacks instead of post-attack mitigations

⊘ **Open Source** — Powered by KubeArmor, 750,000+ downloads

⊘ **Innovative** — 10+ patents, support for IoT/Edge, 5G workloads, AI/LLM

ACCUKNOX

> " LLMs are increasing the availability and variety of attacks by enabling would-be attackers with malicious code development through easy-to-use prompts. GenAI is expected to enhance the efficiency of future cyberattacks by enabling lower-level malicious actors with potentially novel and more advanced attacks"

Gartner 2023

In the era of Generative AI, security cannot be an afterthought. ModelKnox, powered by AccuKnox CNAPP, offers a comprehensive solution for securing your Large Language Models (LLMs) throughout their lifecycle.

This E-book gives you a tour of Key Features from ModelKnox -

- ✓ **End-to-end LLM pipeline security**
- ✓ **Real-time threat detection and remediation**
- ✓ **Compliance with industry standards (OWASP, NIST, MITRE)**
- ✓ **Intuitive dashboard for holistic security posture management**
- ✓ **Securing CUDA workloads (NVIDIA) for enhanced performance and security**

Book a Demo

Don't let security concerns hold back your AI innovations. With ModelKnox, embrace the power of Large Language Models with confidence.