Secure your AI environment as it scales with Kaspersky AI Protect

Enterprise LLM security white paper



kaspersky cybersecurity true to business

Table of contents

1 Executive summary 3

2 Four key data points 4

3 Key takeaways 5

4 Cybersecurity risks across the enterprise AI lifecycle 6

5 How to secure an AI environment 12

6 Value for technology and security leaders 18

7 Secure your AI environment with Kaspersky 20

01

Executive summary



This technical white paper explains how to secure your AI environment before emerging threats can gain a foothold.

You will learn:

- The emerging security risks facing enterprise AI
- How threats can enter AI environments through files, URLs, and data
- How Kaspersky AI Protect (KAIP) protects critical entry points in AI environments
- How to strengthen security as your organization expands its use of AI

Use these insights to build a more secure AI environment.

Intended Audience

This white paper is intended for CISOs, CTOs, heads of AI or AI platforms, and other technical professionals involved in enterprise AI projects in the following sectors:

- Government
- IT
- Finance
- Telecom

Kaspersky's solution for AI environments

Kaspersky AI Protect (KAIP) is designed to establish a robust, multidimensional, full-stack content security layer for enterprise large language model applications.

It is a high-performance security scanning service that provides advanced malware protection, HTTP traffic filtering, and file and URL reputation checks for AI applications.

LEARN MORE

About Kaspersky

Kaspersky is a global cybersecurity company with deep international presence. Currently, we have 34 offices in more than 30 countries and regions, with a team of more than 4,500 experts safeguarding global security. We are committed to innovation-driven development and providing superior, easy-to-use, and efficient security protection. With world-leading technological capabilities, we protect 400 million users and 220,000 enterprises globally. Kaspersky develops innovative, effective, and easy-to-use security solutions that help customers operate safely in a complex and constantly changing threat environment.

02

Four key data points

92,000

malicious attacks disguised as AI services detected by Kaspersky in H1 2026.[1]

10x

increase in recorded vulnerabilities in AI and LLM services between Q4 2025 and Q2 2026 according to Kaspersky Vulnerability Management data.[2]

These findings show that AI infrastructure itself, not only traditional IT, is becoming a target and also needs protection.



32%

of organizations experienced an attack on an enterprise GenAI application. [3]

29%

of organizations experienced an attack on enterprise GenAI application infrastructure. [3]

[1] Kaspersky Detected More than 92,000 Malware Attacks Disguised as AI Services in 2026, (Kaspersky, May 2026)

[2] The Number of Vulnerabilities within AI Services Surged Tenfold in the First Half of 2026, (Kaspersky, August 2026)

[3] Gartner Survey Reveals GenAI Attacks Are on the Rise, (Gartner, September 2025)

03

Key takeaways

1) Threats to AI infrastructure

As organizations adopt AI technologies, their AI infrastructure becomes a target and requires dedicated protection.

Across the AI lifecycle, a wide range of files and data objects are continuously generated and exchanged, including raw corpora, datasets, scripts, dependency packages, container images, model weights, checkpoints, quantized model files, API packages, user-uploaded content, and files for RAG knowledge bases.

Without a unified approach to content security, these assets can become vectors for malicious code, compromised dependencies, backdoors, phishing links, malicious URLs, and tampered model artifacts. This creates significant security risks across the large language model (LLM) development and deployment environment.

2) Reducing cybersecurity risks in AI environments

To establish comprehensive protection against these risks, organizations need to:

- Check data as it enters the environment and neutralize malicious content before it reaches data lakes, object storage, or model training datasets.
- Secure the AI supply chain by scanning training files, artifacts, and container images for malicious code before they are used.
- Select a secure solution that is easy to deploy, scale, and operate, and that integrates seamlessly with third-party solutions and the existing security operations center (SOC).

3) How Kaspersky helps mitigate new AI risks

Drawing on its global threat intelligence and research into adversarial AI defenses, Kaspersky developed KAIP to provide multidimensional, full-stack content security for enterprise LLM applications.

It is the best-in-class solution for detection of AI-borne as well as traditional threats that can be integrated with almost any application.

KAIP provides comprehensive protection for web portals and applications, proxy servers, network attached storage and mail gateways.

KAIP helps organizations avoid substantial financial losses, reduce operating costs, shorten manual audit and incident response times, and accelerate AI adoption.

04

Cybersecurity risks across the enterprise AI lifecycle

Enterprise LLMs are rapidly moving from laboratories into production and are widely used for enterprise knowledge management, office automation and copilots, customer service, and business decision-making. Because LLMs function largely as "black boxes" and depend heavily on unstructured data, they can expose organizations to data breaches, injection attacks, and knowledge base contamination.

Developing and training an LLM is a complex systems-engineering process that typically comprises four core phases, each with distinct risks.



4.1. Data preparation and preprocessing

This stage mainly includes data collection, cleaning, deduplication, sensitive information filtering, tokenization, and vocabulary construction. Data may come from websites, open-source code repositories, academic papers, books, and forums, as well as internal knowledge bases, business systems, and third-party data providers.

Core risks

Malicious files, scripts, URLs, or infected archives from untrusted sources can enter the data lake and training pipeline.



Security consequences

- Compromise of the data-cleaning node
- Malicious samples entering the training dataset
- Lateral movement into the internal data lake
- Contamination of the training corpus
- The model subsequently learning or generating malicious links or code snippets
- Use of the R&D environment as a foothold for further attacks on AI infrastructure

Typical security incident scenario

An organization collects industry data from the internet to train a private LLM, including HTML pages, PDF documents, and archives. Because the data-ingestion point lacks a security scanning mechanism, one archive contains a malicious script disguised as a data sample.
The data-cleaning node executes the script during automated decompression and parsing, installing a backdoor on the server. The attacker then uses the compromised node to access the data lake and steal internal corpora, training samples, and business documents.

4.2. Pretraining stage

This stage uses massive text datasets for unsupervised learning and is typically the most computationally intensive and time-consuming phase of LLM training. It involves training code and scripts, dependency packages, container images, distributed training frameworks, data shards, checkpoints, tokenizers, vocabulary files, and logs.

Core risks

Contamination of the training supply chain or training artifacts can lead to malicious code executing in the training cluster or to the creation of untrusted model artifacts.



Security consequences

- Backdoors implanted on training-cluster nodes
- GPU resources hijacked for crypto-mining or other unauthorized use
- Exposure of training data and model weights
- Contaminated checkpoints compromising subsequent training
- Tampered model artifacts entering downstream fine-tuning or deployment processes
- Supply-chain compromise spreading across the MLOps platform

Typical security incident scenario

To accelerate training, an LLM team temporarily downloads a data-loading toolkit from an external source.
The package name closely resembles that of a well-known open-source component, indicating a dependency-confusion attack.
When the training image is built, the package is installed automatically, and its installation script executes malicious code that steals credentials for the image repository and access keys for object storage.

The attacker then uses the credentials to access training data and checkpoints and attempts to upload tampered model files.

4.3. Fine-tuning and alignment stage

The fine-tuning and alignment stage includes supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF), and direct preference optimization (DPO). These processes are intended to help the base model follow human instructions and produce outputs that meet business and security requirements and align with organizational values. This stage uses prompt-response pairs, preference data, manually annotated data, business knowledge, conversation logs, security policy examples,
and red-team test samples.

Core risks

Malicious content can be embedded in labeled datasets, human feedback, business knowledge files, or lightweight model adaptation files, compromising the fine-tuning environment and the integrity of its data.



Security consequences

- Infection of the annotator's endpoint
- The annotation platform becoming a channel for distributing malicious files
- Exposure of fine-tuning data and business knowledge
- Malicious URLs entering the prompt-response dataset
- Contamination of LoRA, adapter, and other fine-tuning artifacts
- Uncontrolled spread of red-team samples into the general R&D environment

Typical security incident scenario

An organization fine-tunes a customer service model by importing historical support tickets, email attachments, and customer service conversation logs into an annotation platform. Some tickets contain user-uploaded archives and Microsoft Office attachments. One Excel file contains a malicious macro. When an annotator downloads and opens the attachment during a sample review, the macro executes and gives the attacker remote access to the endpoint.

The attacker then uses the compromised endpoint to access the annotation platform and download fine-tuning data and business knowledge samples in bulk.

4.4. Evaluation, quantization, and deployment stage

The evaluation, quantization, and deployment stage includes benchmark testing, model compression, model quantization, inference service encapsulation, API release, container image creation, plugin integration, RAG knowledge base construction, production operation, and user access. At this stage, the model moves from the R&D environment into production.

Core risks

Malicious content can be embedded in model artifacts, deployment packages, user-uploaded files, RAG knowledge base files, or API traffic, directly affecting production security.



Security consequences

- Malicious files compromising production AI applications
- Compromise of document-parsing nodes
- Contamination of the RAG knowledge base
- The user-upload channel becoming a distribution point for malicious files
- Malicious URLs appearing in model responses
- Abuse of production API services and damage to business credibility
- Widespread infection of internal user endpoints

Typical security incident scenario

An organization deploys a document question-answering bot that allows employees to upload PDF, Word, and Excel files, as well as archives, for analysis. Using a compromised account, an attacker uploads a Microsoft Office document containing malicious macros. Because files are not scanned at the upload point, the document-parsing service processes the file automatically and synchronizes it with the RAG knowledge base. Several employees subsequently download and open the original attachment from the knowledge base, causing the malicious macros to execute on multiple devices. At the same time, the model includes a malicious URL from the document in its responses, further increasing the impact.

4.5. AI lifecycle risk summary

Throughout the AI lifecycle, many files and data objects are continuously generated and exchanged, including raw corpora, cleaned training datasets, training scripts, dependency packages, container images, model weights, checkpoints, quantized model files, API deployment packages, user-uploaded content, and files for RAG knowledge bases.

Without a unified mechanism for detecting malicious content, these files and objects can introduce malicious code, compromised dependencies, backdoors, phishing links, malicious URLs, and tampered model artifacts into the LLM development environment.



05

How to secure an AI environment

5.1. Overview of Kaspersky AI Protect

Traditional cybersecurity architectures cannot always analyze and block attack vectors specific to AI environments. Drawing on its global threat intelligence and research into adversarial AI defenses, Kaspersky developed Kaspersky AI Protect (KAIP) to provide multidimensional, full-stack content security for enterprise LLM applications.

KAIP is a high-performance security scanning service that provides advanced malware protection, HTTP traffic filtering, and file and URL reputation checks for AI applications.

5.2.Product use cases

1) Supply-chain security: Scan model artifacts before they are stored, shared, or deployed to detect malicious code and deserialization risks
2) Infrastructure protection: Integrate scanning into CI/CD pipelines, model repositories, AI gateways, and other security checkpoints to prevent compromised models from reaching production
3) Secure model onboarding: Scan models from external or third-party sources before admitting them to internal AI environments, helping organizations verify that downloaded or shared model artifacts are safe to use
4) Kaspersky AI Protect (KAIP) can serve as a unified malicious content detection engine within an LLM development platform. It can be embedded in data pipelines, MLOps platforms, CI/CD pipelines, artifact repositories, object storage, RAG knowledge base ingestion points, and API gateways to scan files, URLs, and HTTP traffic



5.3. Deployment options

KAIP supports Windows and Linux environments and offers flexible integration options for different business requirements:

1) HTTP (REST-like) mode: KAIP receives client requests through a RESTful service interface, scans transmitted objects efficiently, and immediately returns the results. This mode supports tight integration with custom applications
2) ICAP mode: KAIP operates as a standard ICAP server and scans traffic for proxy servers, network-attached storage (NAS), web application firewalls (WAFs), and next-generation firewalls (NGFWs). It can filter URLs requested by users and automatically block malicious, phishing, or adware-hosting webpages

5.4. Features

1) Scans widely used LLM model-weight file formats, including .pth, .pt, .pkl, .keras, and .ckpt
2) Scans AI agent skill files
3) Advanced detection: Kaspersky's anti-malware technology delivers high detection rates and can respond to emerging threats within milliseconds
4) Intelligent analysis engine: Advanced heuristic analysis and machine learning detect objects packed in multiple layers and identify threat variants
5) Deep cleansing and repair: Disinfects infected files, archives, and encrypted objects while preserving the integrity of the original data whenever possible
6) Cloud reputation intelligence: Uses Kaspersky Security Network (KSN) to obtain real-time reputation data for files and webpages, improving detection speed and accuracy
7) Elasticity and scalability: KAIP supports clustered deployments. Organizations can run multiple KAIP instances on the same network, manage them centrally through a unified web interface, and scale capacity as needed



5.5. KAIP across the AI lifecycle

1) Data preparation and preprocessing

KAIP can detect malicious content before data enters a data lake, object storage, or a training dataset, establishing a security checkpoint at data ingestion.

Recommended deployment methods:

- Call the KAIP HTTP/REST-like interface from the data acquisition platform
- Create temporary and production areas at the object-storage ingestion point; scan data before moving it into production storage
- For data downloaded over HTTP, integrate KAIP at the proxy layer through ICAP
- Deploy KAIP as a unified scanning service in Kubernetes for use by data pipelines

Benefits at this stage:

- Protect annotation platforms and endpoints
- Filter malicious URLs and attachments
- Ensure that tuning data and LoRA or adapter artifacts are free from contamination, supporting safe and reliable model training and release

2) Pretraining stage

KAIP can detect malicious code in training-related files and artifacts before training starts, during container image builds, and when artifacts are uploaded, establishing secure access controls for the training supply chain.

Recommended deployment methods:

- Add a KAIP scanning step to the CI/CD pipeline
- Call KAIP at the artifact repository's upload point
- Scan image layers before publishing container images to the repository
- Scan training packages and configuration files before MLOps tasks begin
- Scan checkpoints and model files before adding them to the model registry

Benefits at this stage:

- Prevent supply-chain poisoning
- Block the spread of malicious images
- Protect checkpoints and model artifacts

3) Fine-tuning and alignment stage

KAIP can integrate with annotation platforms, dataset management platforms, URL extraction services, and red-team sample libraries to check files and URLs as they enter the fine-tuning and alignment environment.

Recommended deployment methods:

- Call KAIP from the annotation platform's file-upload interface
- Add a "scan status" field to the dataset management system
- Extract URLs from prompts and conversation logs, then submit them to KAIP for scanning
- Establish isolated zones and controlled access processes for the red-team sample library
- Scan LoRA, adapter, and model-weight files generated during fine-tuning before release

Benefits at this stage:

- Protect annotation platforms and endpoints
- Filter malicious URLs and attachments
- Ensure the integrity of tuning data and LoRA artifacts

4) Evaluation, quantization, and deployment stage

KAIP can protect the model both before and after deployment.

Before deployment:

- Scan model weights, quantization files, tokenizers, and configuration files
- Scan inference service images, deployment packages, and startup scripts
- Block infected artifacts from entering the production environment

After deployment:

- Scan user-uploaded files
- Scan files imported into RAG knowledge bases
- Inspect HTTP traffic and URLs
- Send alerts about malicious files and URLs to the SOC platform
- Reject, isolate, delete, or route detected content for manual review

Recommended deployment methods:

- Call KAIP from the annotation platform's file-upload interface
- Add a "scan status" field to the dataset management system
- Extract URLs from prompts and conversation logs, then submit them to KAIP for scanning
- Establish isolated zones and controlled access processes for the red-team sample library
- Scan LoRA, adapter, and model-weight files generated during fine-tuning before release

Benefits at this stage:

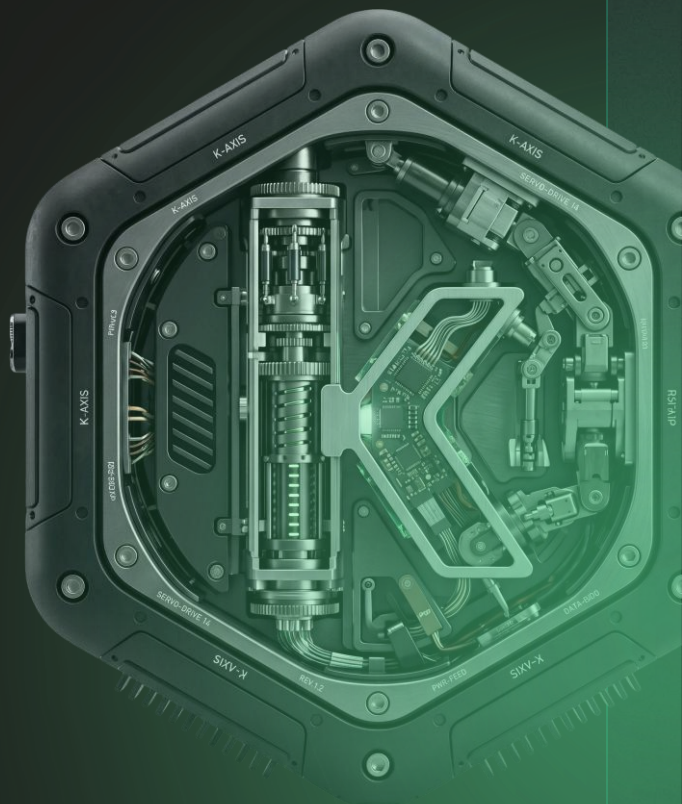
- Protect the production deployment chain
- Prevent contamination of RAG knowledge bases
- Block malicious content from entering applications

Overall security architecture summary

Integrating KAIP throughout the LLM lifecycle creates a unified content security scanning layer. All data, files, model artifacts, and user inputs should be scanned before entering critical stages of the AI lifecycle.

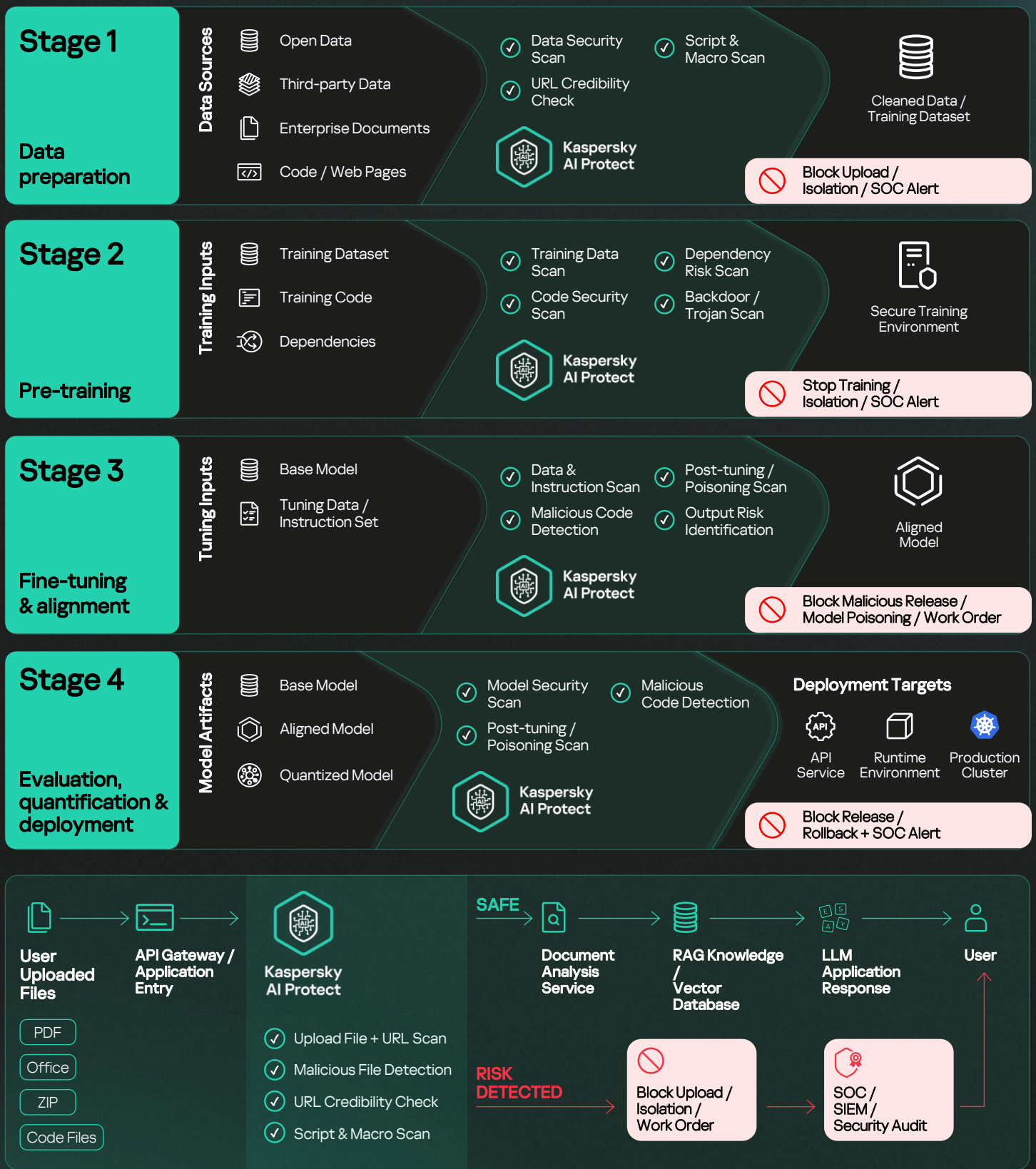
KAIP does not replace model security evaluation. Instead, it provides foundational malicious-content detection across the LLM development and deployment chain. It protects the following entry points:

- Data ingestion
- Training supply chain entry point
- Fine-tuning and annotation entry point
- Model artifact release entry point
- RAG knowledge base entry point
- User uploads to production applications
- HTTP and URL access entry point



Kaspersky AI Protect (KAIP) overall security architecture

Complete lifecycle of large model development



5.6. General deployment architecture

The architecture comprises the kavhttpd HTTP service, Kaspersky Anti-Virus Engine, and the solution's web graphical user interface.

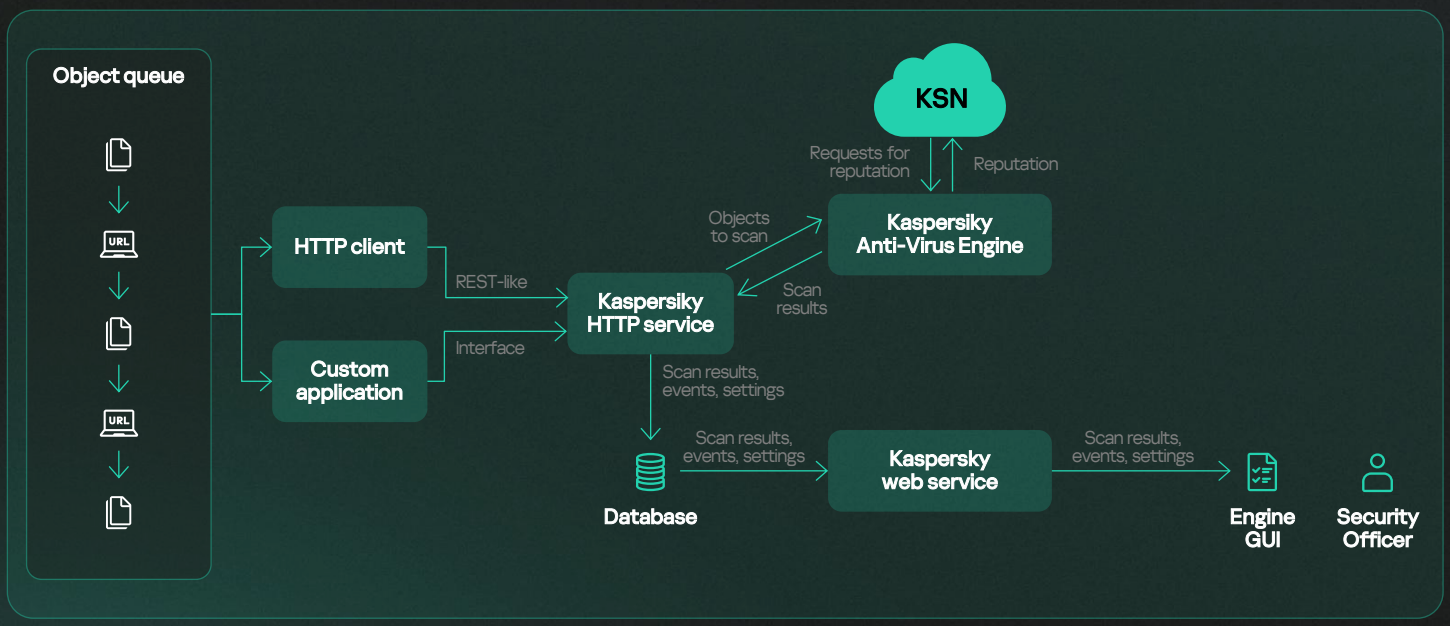
When Kaspersky AI Protect operates in HTTP mode, the components interact in the following order:

1) Requests are sent to kavhttpd over HTTP
2) The kavhttpd service sends the objects to Kaspersky Anti-Virus Engine
3) Kaspersky Anti-Virus Engine scans the objects

   When file and URL reputation checks are enabled, the objects are also sent to KSN for reputation checks.
4) After scanning is complete, Kaspersky Anti-Virus Engine returns the results to the kavhttpd service
5) The kavhttpd service sends the scan results to the HTTP clients or other applications that submitted the objects

   When the web GUI is used, the results appear on the scan results page.

   The following diagram shows how the product components interact.



06

Value for technology and security leaders

KAIP is designed for senior technology and security leaders in government, financial services, IT, and telecommunications.

Their responsibilities and priorities include:

CISO

Oversees the organization's cybersecurity and risk management, with a focus on reducing security and supply-chain risks related to AI.

CTO

Oversees technology strategy and AI adoption, with a focus on integrating AI security into the existing infrastructure.

Head of AI

Oversees AI and machine learning platforms and the model lifecycle, with a focus on protecting model repositories and deployment pipelines.

Kaspersky AI Protect delivers value across six pillars:

6.1. Zero Trust security for AI

KAIP applies Kaspersky's Zero Trust approach to AI security. All inputs and outputs are treated as untrusted, regardless of whether the data comes from internal users, trusted partners, or the AI model itself. KAIP performs real-time security and compliance checks at every stage of the interaction.

6.2. Unified data protection across AI entry points

KAIP controls data entering the AI ecosystem at four key points:

File uploads: When users upload individual files through a chat interface or import attachments in batches through a business system, KAIP applies anti-malware and heuristic scanning to detect hidden malicious payloads and adversarial samples.

RAG knowledge base ingestion: Before documents are vectorized and embedded, KAIP scans source data to prevent malicious scripts, contaminated data, and text containing prompt-injection instructions from entering the vector database.

API and plugin calls: KAIP monitors requests made by AI agents to external systems. It inspects request parameters and returned content to help prevent server-side request forgery (SSRF), malicious command execution, and sensitive data leakage.

Internal enterprise data flows: KAIP integrates with enterprise email, cloud storage services such as Nextcloud and OneDrive, and collaboration platforms such as Slack and Teams to prevent external threats from reaching AI systems through these channels.

6.3. Flexible deployment architecture

KAIP uses a cloud-native, modular architecture designed for the concurrency, latency, data-residency, and compliance requirements of medium and large enterprises.

- High-availability cluster deployment: KAIP supports deployment on internal Kubernetes (K8s) clusters or standalone security-service clusters, with automatic scaling based on traffic.
- Flexible access protocols: Business systems can call KAIP through high-concurrency RESTful APIs or connect it to an enterprise network gateway through the Internet Content Adaptation Protocol (ICAP) for transparent traffic inspection.
- Performance: Kaspersky's streaming scanning and in-memory unpacking technologies keep large-file scanning latency in the millisecond range, minimizing the effect on the AI system's time to first token (TTFT).

6.4. Integration with existing security operations centers

- Standardized log delivery: KAIP can convert scan events and blocked-threat records to Syslog or Common Event Format (CEF) and send them to SIEM platforms such as Splunk or QRadar in real time.
- SOAR playbook integration: When KAIP detects a high-risk keyword injection or RAG poisoning event, it can trigger a Security Orchestration, Automation, and Response (SOAR) playbook. The playbook can temporarily suspend the suspected account in the identity and access management (IAM) system, block the malicious IP address at the firewall, and send a high-priority alert to the security team, enabling a rapid response.

6.5. AI compliance and governance

AI regulations are becoming more stringent, and KAIP can support compliance efforts.

- NIST AI RMF (Artificial Intelligence Risk Management Framework): KAIP provides detailed audit logs for AI inputs and outputs, supporting the framework's trustworthiness, security, and governance objectives.
- ISO/IEC 42001 (Artificial Intelligence Management System): KAIP provides technical controls and data that can help organizations implement AI governance, conduct traceable audits, and meet certification requirements.

6.6. Measurable business value and ROI

Introducing KAIP can provide measurable business benefits in addition to improving security.

- Avoid financial losses: KAIP helps prevent losses and regulatory penalties associated with the exposure of intellectual property and financial data.
- Reduce operating costs: Automated scanning and blocking reduce the volume of unknown AI-related threats that Security Operations (SecOps) teams must analyze manually, shortening audit and incident response times.
- Accelerate AI adoption: Stronger security controls can address concerns from management and compliance teams, allowing organizations to implement AI initiatives more confidently and quickly.

It's important to point out that KAIP primarily addresses conventional cybersecurity threats, including malicious files, URLs, scripts, Trojans, backdoors, ransomware, and phishing content. AI-native threats, including prompt injection, model jailbreaking, semantic-layer data poisoning, model bias, and hallucinations, require complementary controls such as data governance, model security evaluations, red-team testing, content-security policies, and runtime protection.

As an AI data security gateway, KAIP provides unified deep file scanning, malicious URL detection, and bidirectional analysis of API interactions. It creates an isolation and content-cleansing boundary between enterprise users, business systems, and LLMs, helping enterprise AI applications operate securely and reliably while meeting compliance requirements.



07

Secure your AI environment with Kaspersky



Contact us

to build security into your organization's AI journey, from development and training to deployment and everyday use by employees.