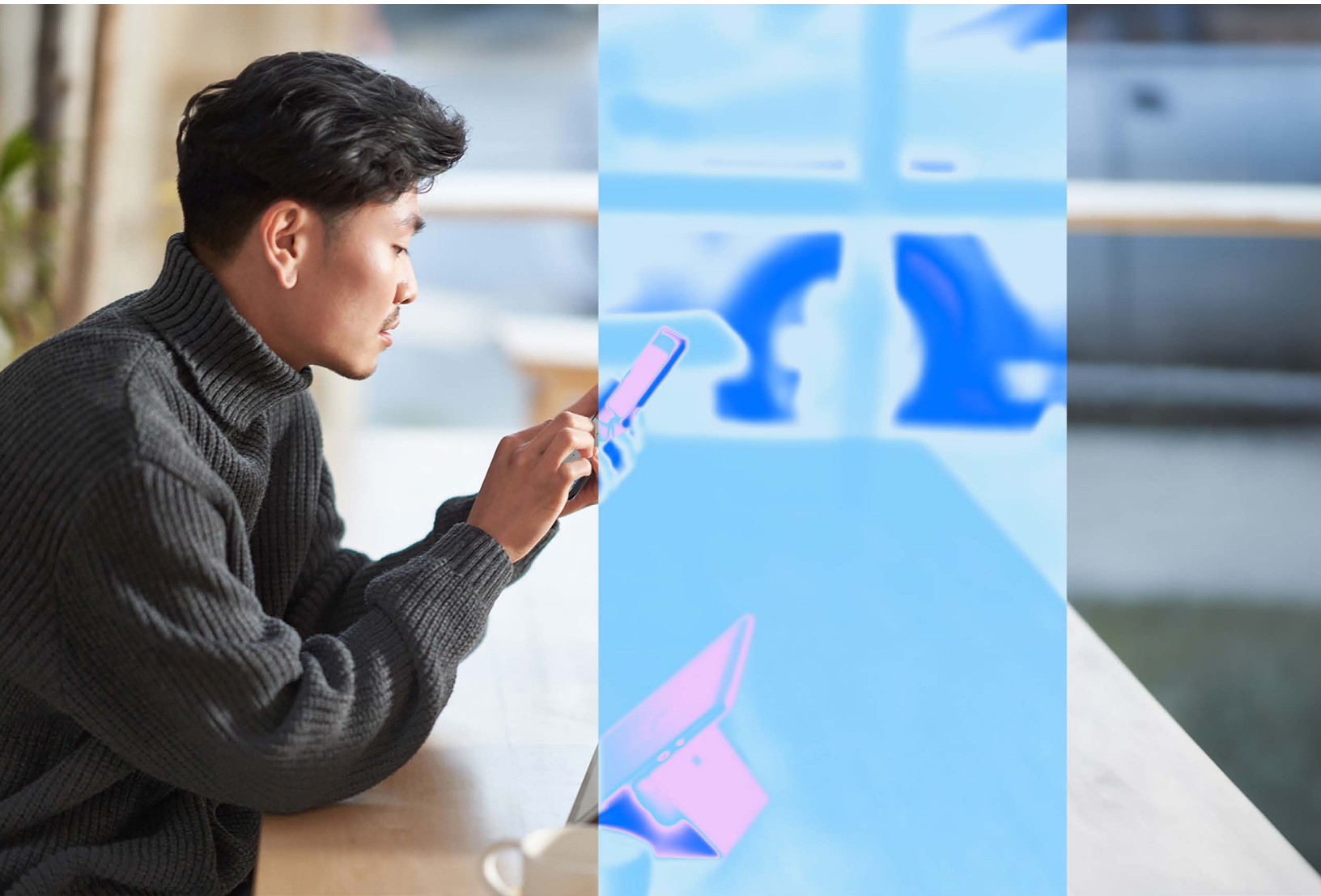
Microsoft Security

E-book

Stopping Generative AI Threats in Runtime:

5 Common Attack Vectors and How to Remediate Them



Contents

Introduction

We are in the midst of a once-in-a-lifetime AI revolution. Today, nearly three-quarters (72%) of companies have integrated AI into at least one business function, and 65% have embraced generative AI specifically.[1]

However, just as businesses are using AI to improve operations, threat actors are using the technology to improve the speed and sophistication of their attacks. According to Gartner, AI-enhanced malicious attacks and AI-assisted misinformation were the top two most common emerging enterprise risks in 2024.[2] Among ethical hackers, 77% use AI technology as part of their security research workflows.[3]

At the same time, organizations are building custom AI applications and grounding commercially available models in proprietary data through retrieval augmented generation (RAG). Nearly half (47%) of companies significantly customized existing models or developed proprietary models in 2024.[4] While these tactics can help improve the accuracy and relevancy of AI-generated results, they also open organizations up to increased risk. Not only must today's security teams defend against new types of generative AI-specific threats, but they also have to consider risk factors like potential misconfigurations or improper data governance controls.

Ethical Hacker

Also known as white hat hackers, these security researchers are hired by organizations to penetrate computing resources and uncover potential vulnerabilities that malicious hackers could exploit.

Retrieval Augmented Generation (RAG)

This process augments the capabilities of large language models (LLMs) by adding an information retrieval system that uses custom or proprietary grounding data to improve the accuracy and relevancy of AI-generated responses.

Key Challenges Surrounding AI Security at Runtime

Applications are cloud-based: Generative AI applications are typically cloud-based and do not rely on self-developed LLM models. In fact, the technology is estimated to have contributed to half of the increase in cloud service revenues last year.[5] Attackers can exploit generative AI's cloud-based nature by leveraging vulnerabilities within the model, application, or other areas of the cloud environment to move laterally and compromise sensitive training or user data.

Models require access to large datasets: Generative AI also requires access to large amounts of structured and unstructured data. Not only does this make it an attractive target for threat actors, but it's also difficult for security teams to protect the sheer scale of data that the technology uses. By 2027, more than 40% of AI-related data breaches will be caused by the improper use of generative AI across borders.[6] This trend highlights the need for consistent AI best practices and data governance standards across organizations' digital estates.

Model outputs and behavior are non-deterministic: Finally, generative AI is non-deterministic, making it difficult for security teams to accurately predict and control the model's behavior. Attackers can exploit this non-deterministic nature to manipulate model outputs or gain unauthorized access to sensitive data or cloud environments, as seen in attack vectors like jailbreaks and prompt injections—which The National Institute of Standards and Technology (NIST) has described as generative AI's "greatest security flaw."[7] By 2028, 25% of enterprise breaches will be traced back to AI agent abuse by external and malicious internal actors—highlighting the need for organizations to implement mitigation strategies to protect against AI-specific risk.[8]

To combat these and other security risks, organizations must take a broader view of runtime security for AI applications. In the same way that organizations have built rigorous cloud security programs, they need a similar strategy for generative AI security. Rather than narrowly focusing on blocking individual attack vectors, organizations must take a holistic approach that unifies cloud-native posture management with threat protection.

1 "AI adoption among organizations worldwide 2017-2024, by type," Statista.
2 "Gartner Survey Shows AI Enhanced Malicious Attacks as Top Emerging Risk for Enterprises for Third Consecutive Quarter," Gartner.
3 "Inside the mind of a hacker 2024," Bugcrowd.
4 "The state of AI: How organizations are rewiring to capture value," McKinsey.

5 "Gen AI Re-Accelerates Cloud Market Growth," Statista.
6 "Gartner Predicts 40% of AI Data Breaches Will Arise from Cross-Border GenAI Misuse by 2027," Gartner.
7 "Indirect Prompt Injection: Generative AI's Greatest Security Flaw," Centre for Emerging Technology and Security.
8 "Gartner Unveils Top Predictions for IT Organizations and Users in 2025 and Beyond," Gartner.

Benefits of Unified Cloud-Native Security Controls for Generative AI

There are two main approaches companies can take when securing their cloud environment: best-of-breed or cloud-native. Best-of-breed allows companies to select individual tools across multiple vendors that excel in specific areas of cloud security, while cloud-native embeds security directly into the design and operation of cloud systems.

The challenge with best-of-breed is that it opens companies up to increased complexity, larger attack surfaces, and security gaps for cloud workloads. Third-party cloud security solutions rely on the cloud service provider's (CSP's) application programming interface (API) for visibility. Each new tool also introduces its own set of configurations, APIs, and potential vulnerabilities that must be monitored and maintained. These third-party solutions often struggle to integrate with one another and with the cloud platform, which can lead to visibility gaps and prevent teams from establishing a unified view of their security landscape.

At Microsoft, we natively integrate Azure OpenAI Service with our cloud-native application protection platform (CNAPP), Microsoft Defender for Cloud. This allows us to provide full-lifecycle security for generative AI applications and seamlessly correlate security alerts across multiple domains. Because our native-first approach relies on first-party solutions, customers don't have to make any changes to their cloud environment or manually integrate tools. All customers who deploy Defender for Cloud automatically have access to Azure OpenAI as part of their subscription.

Cloud-native application protection platform: A unified platform that simplifies cloud-native application and infrastructure security by integrating multiple solutions to embed security from application development to provisioning and runtime, helping to mitigate risks across hybrid and multicloud environments.

An integrated approach delivers better runtime security for generative AI.

Secure generative AI applications from development to deployment: Because Azure OpenAI natively integrates with Defender for Cloud, security best practices are enforced across the full application lifecycle. Defender for Cloud offers a development security operations (DevSecOps) solution that unifies security management at the code level. Meanwhile, AI security posture management (AI-SPM) automatically and continuously discovers deployed AI workloads, which are then monitored for malicious activity using threat protection.

Streamline development with workload transparency: When organizations take a non-native approach to application security, developers must configure application workloads with a man-in-the-middle (such as a proxy server) to inspect all requests before the application can call on the AI model. This allows teams to verify whether or not the request complies with the organization's security policies. However, in a native approach, the AI model is integrated directly into the cloud platform's backend rather than connecting to the application itself. For organizations deploying Defender for Cloud (which natively integrates with Azure OpenAI), security features are built into the service, so development teams don't need to make changes to the application's workload to enforce security. Instead, Defender for Cloud secures everything from the application's workload performance to API management, data storage, access permissions, and more.

Maximize your existing investment: Likewise, for organizations that are using Azure OpenAI, Defender for Cloud security features are already bundled into the price. Rather than implementing and learning a new security system to protect generative AI workloads, teams can simply maximize their existing investment by deploying Defender for Cloud's unified cloud-native approach.

Deliver comprehensive security controls: Azure OpenAI's native features provide a wide range of security controls that help protect AI workloads from various threats. For example, Defender for Cloud can use its native integration with Azure OpenAI and Azure AI Content Safety Prompt Shields to provide contextual and actionable security alerts about harmful user- or AI-generated content in applications and services.

Centralize application management: As a CNAPP, Defender for Cloud integrates solutions like Azure API Management and Microsoft Entra ID (our identity and access management solution) under a single umbrella. This allows organizations to centralize authentication, authorization, and traffic control to prevent unauthorized users or applications from interacting with their AI models. Rather than asking security teams to manually correlate alerts across separate solutions, Defender for Cloud automatically contextualizes insights to help teams understand which risks they need to address first.

Ensure compliance and data governance: Azure native solutions help maintain compliance with industry standards and regulations. For instance, our data platform, Microsoft Fabric, is HIPAA compliant—ensuring the best level of security, compliance, and privacy for customer data.

Integrate with existing tools: Azure native features also integrate seamlessly with other Microsoft tools and services, such as GitHub, Visual Studio, and Copilot Studio. This makes it easier for teams to manage and secure their generative AI applications.

Take a holistic approach to application security: By addressing multiple aspects of application security within a single platform—including configuration, operation, and data security—Azure native solutions provide a more holistic approach to securing AI workloads.

Enforce security best practices: Because Defender for Cloud unifies multiple security solutions within a single pane of glass, it can provide detailed security guidance for AI workloads on both platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) cloud service models. This allows teams to seamlessly secure Azure AI models, resources, and data.

Implement access control and audit logging: Finally, organizations can ensure their AI workloads are secure and industry-compliant by using Azure OpenAI policies to implement access control, audit logging, network isolation, and encryption.

Top 5 Generative AI Threats in Runtime

So, now that we've explored the benefits of taking a unified cloud-native approach to securing generative AI applications, what are the top threats security teams should guard against in runtime? The following list of attack vectors is based on OWASP's top risks for LLM applications in 2025 and the MITRE ATLAS Matrix.[9, 10]

1. Poisoning attack: Poisoning attacks attempt to manipulate or corrupt the training data of AI and machine learning (ML) models to influence their behavior and potentially compromise their accuracy, reliability, or ethical responsibility. Attackers will either inject false or misleading information, modify, or delete portions of the training dataset to change how the model responds to user queries. This can be done by targeting the cloud storage accounts where training data is hosted.

2. Evasion attack: Evasion attacks circumvent existing AI security systems by modifying or manipulating input data in a way that the model cannot detect. For example, attackers might hide spam content within an image (thus bypassing the model's spam filters) or use network packet fragmentation to trick the model's intrusion detection system (IDS). Evasion attacks can be targeted, meaning the attacker is trying to create a specific error in the model's output, or untargeted, meaning that the attacker is trying to get the model to output any type of error. One commonly understood type of evasion attack is AI model jailbreaks.

3. Functional extraction: In functional extraction attacks, adversaries repeatedly query AI models and use the responses to recreate and train functionally equivalent models. Attackers can then analyze this substitute model before launching further attacks on the commercially available version.

4. Inversion attack: Similar to functional extraction, during inversion attacks, threat actors repeatedly query AI models and use the outputs to infer information about the model's parameters or architecture. This type of attack can also allow adversaries to extract sensitive information about the model's training data, including everything from complete training data reconstruction to insights about specific data attributes or properties. Inversion attacks can be effective on their own or serve as a precursor to other threats, such as evasion attacks.

5. Prompt injection attack: Just as the name describes, prompt injection attacks use malicious AI prompts to manipulate the model into behaving in unintended or harmful ways. These prompts are typically designed to make the model disregard its original instructions and follow the attacker's commands instead. Prompt injections are rooted in generative AI's non-deterministic nature, making them difficult to prevent.

9 "OWASP Top 10 for LLM Applications 2025," OWASP.
10 "ATLAS Matrix," MITRE.

Threat Remediation Tactics for Generative AI Workloads

Based on the attack vectors described above, the following provides a brief overview of how organizations can use Defender for Cloud to protect their generative AI applications in runtime.

Poisoning Attacks

While attackers may not have access to the AI model itself, there are several strategies they can use to compromise its training data. To protect generative AI applications against poisoning attacks, security teams should:

• Follow Azure best practices for generative AI, including data segregation, identity and access management, network security, application security, and governance.

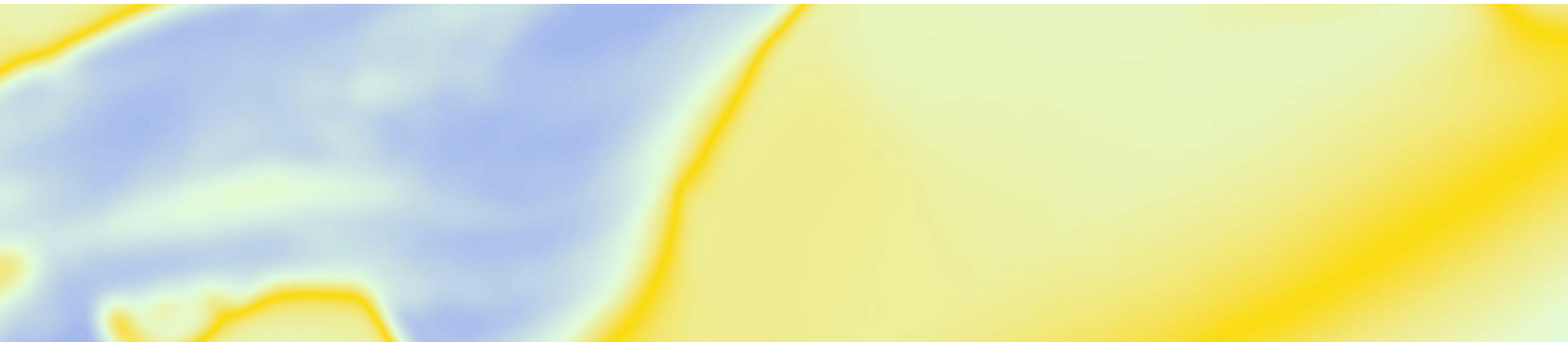
• Leverage threat protection for AI services in Defender for Cloud to detect potential poisoning attacks through alerts like data exposure, credential theft attempts, and phishing.

• Before pushing generative AI applications into production, use data validation to proactively identify risks and assess the model's outputs for accuracy, groundedness, and relevance.

• Educate users about the importance of evaluating model outputs, including checking AI-generated images for Content Credentials. These Content Credentials use open technical specification from the Coalition for Content Provenance and Authenticity (C2PA) to provide a tamper-evident way for disclosing the origin and history of content.

• Leverage threat protection for AI services in Defender for Cloud, including abuse monitoring capabilities, to detect when generative AI applications are being misused and mitigate further abuse. Abuse monitoring leverages content classification, abuse pattern capture, automated and human reviews, and email notification to detect and mitigate recurring content or service abuse.

Learn more about poisoning attacks >



Evasion Attacks

Because evasion attacks are designed to mislead AI models during the inference stage, attackers often use them as a stealthy and effective way to bypass inherent security controls in the AI service. To protect generative AI applications against evasion attacks, security teams should:

• Continuously analyze and monitor LLM inputs by deploying Azure AI Content Safety Prompt Shields in Azure OpenAI. Prompt Shields act as a unified API that analyzes LLM inputs and detects adversarial user input attacks.

• Detect and prevent the output of harmful content by deploying content filtering within Azure OpenAI. This content filtering system uses an ensemble of classification models to detect and act on specific categories of potentially harmful content in both input prompts and output completions.

• Use Defender for Cloud to detect jailbreak attacks on AI workloads. Jailbreak alerts are designed to notify the SOC any time there is an attempt to manipulate system prompts to bypass generative AI's safeguards, potentially accessing sensitive data or privileged functions.

• Leverage threat protection for AI services in Defender for Cloud, including abuse monitoring capabilities, to detect when generative AI applications are being misused and mitigate further abuse. Learn more about abuse monitoring in the Resources section.

• Explore the processes, techniques, and tools that Microsoft's AI Red Team (AIRT) uses to evaluate high-risk AI and identify vulnerabilities in pre-ship products, including PyRIT AIRT's open-source automation framework.

Learn more about evasion attacks >

Functional Extraction and Inversion Attacks

Because functional extraction and inversion attacks take a similar approach in which adversaries will repeatedly query the model for nefarious purposes, organizations can use similar techniques to mitigate both types of attacks. Typically, adversaries are unable to access the model directly, so there are multiple places in the AI workflow where security teams can implement controls.

To protect generative AI applications against functional extraction and inversion attacks, security teams should:

• Use rate limiting to restrict the number of API calls a user or service can make in a given time frame. This ensures fair usage and prevents any single user or service from monopolizing API resources.

• Detect and prevent the output of harmful content by deploying content filtering within Azure OpenAI.

• Leverage threat protection for AI services in Defender for Cloud, including abuse monitoring capabilities, to detect when generative AI applications are being misused and mitigate further abuse.

• Use Content Credentials in Azure OpenAI to identify if the model has been copied or extracted. This can serve as a deterrent for attackers and provide evidence of unauthorized use in the event of a breach.

Learn more about functional extraction and inversion attacks >

Prompt Injection Attacks

Unlike attacks that require detailed knowledge of an AI model's internal workings, prompt injections leverage the model's inherent sensitivity to how prompts are phrased and structured. Because generative AI is non-deterministic, it's difficult for security teams to fully predict all the ways in which an attacker might structure a prompt to produce harmful outputs. This makes prompt injection both effective and adaptable across various applications, including chatbots and other systems powered by LLMs. Prompt injections can also exploit AI's inherent inability to separate data from code by embedding malicious instructions inside objects to bypass system prompts and safety guards.

To protect generative AI applications against prompt injections, security teams should:

• Use Microsoft's Python Risk Identification Tool (PyRIT) to test generative AI workloads and proactively identify system risks before application deployment. PyRIT is an open source framework designed to empower security professionals and engineers to proactively identify risks in generative AI systems.

• Continuously analyze and monitor LLM inputs by deploying Azure AI Content Safety Prompt Shields in Azure OpenAI.

• Detect and prevent the output of harmful content by deploying content filtering within Azure OpenAI. Learn more about content filtering in the Resources section.

• Implement input validation to ensure that only properly formed data enters the AI workflow. When applied at the syntactic and semantic levels, input validation can prevent malformed data from persisting in the database and causing malfunction in downstream components.

• Use threat protection for AI services within Defender for Cloud to detect prompt injection attacks on AI workloads.

• Leverage threat protection for AI services in Defender for Cloud, including abuse monitoring capabilities, to detect when generative AI applications are being misused and mitigate further abuse.

Learn more about prompt injection attacks >

Best Practices for Enhancing Generative AI Security

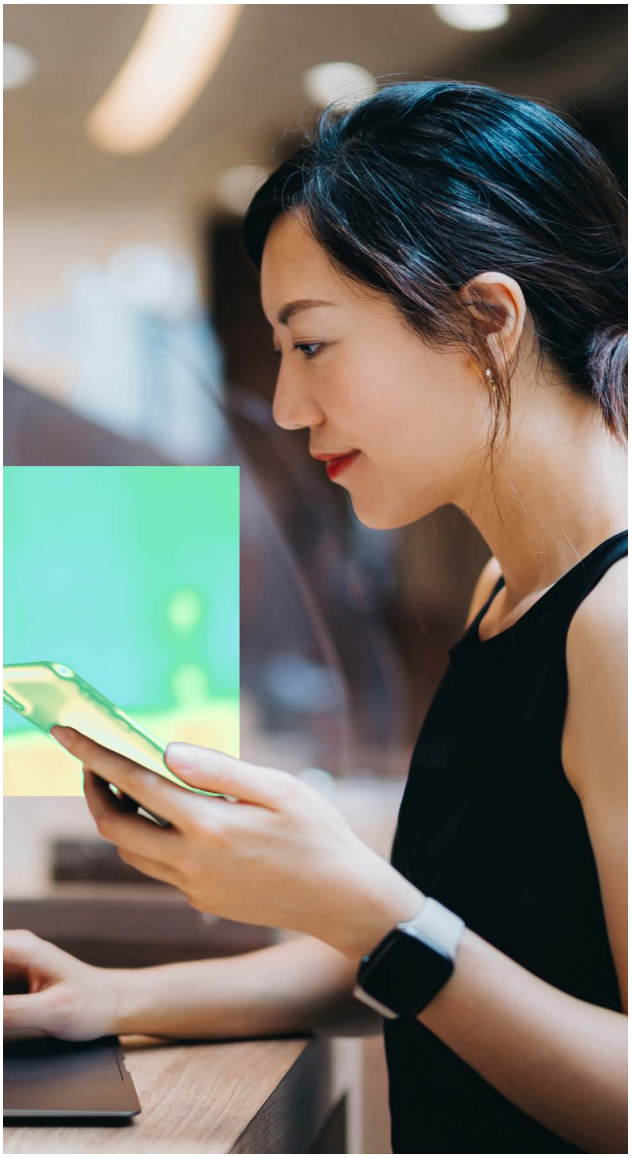
In addition to focusing on threat protection, organizations must also take a more holistic approach to generative AI security that incorporates cloud-native posture management. Implementing this kind of comprehensive security program can be broken down into three core phases.

Phase 1: Assess

Organizations should start by leveraging AI-SPM within Defender for Cloud to:

- Discover AI workloads deployed in their environment using Cloud Security Explorer.
- Identify all code repositories within the environment that contain known generative AI vulnerabilities and provision Azure OpenAI.
- Validate if generative AI workloads deviate from the baseline.
- Review security issues (also known as attack paths) that pose immediate threats and have the greatest potential for exploitation for generative AI workloads.
- detecting vulnerabilities within generative AI library dependencies, and scanning source code for Infrastructure as Code (IaC) misconfigurations and container images for vulnerabilities.
- Use the AI Risk Database to review the risks associated with public ML models.

Find best practice resources for the assess phase >



Phase 2: Plan

- Configure content filters in Azure OpenAI Service to detect and prevent harmful input prompts and output completions.
- Decide which logging and monitoring mechanism the organization will use for Azure OpenAI Services. Since it's difficult for security teams to fully detect prompt injections, organizations can also use prompt logs to validate whether sensitive data was revealed by the model or if the model's training data was poisoned.
- Enable threat protection for AI services in Defender for Cloud for runtime detection and remediation. This feature is designed to protect AI workloads by providing insights to threats that might affect generative AI applications.
- Review common security alerts for generative AI workloads in Defender for Cloud.
- Upskill the security operations center (SOC) team on how to investigate and manage generative AI incidents.

Find best practice resources for the plan phase >

Phase 3: Implement

- Enable Azure Policy to detect any Azure OpenAI instances without diagnostic logging (use Policy ID: /providers/Microsoft.Authorization/policyDefinitions/1b4d1c4e-934c-4703-944c-27c82c06bebb). Azure Policy is designed to enforce organizational standards and assess compliance at-scale by providing an aggregated view of all Azure resources within a unified compliance dashboard.
- Ensure generative AI logs (whether diagnostic or Azure API Management) are being captured in Microsoft Sentinel, our cloud-native security information and event management (SIEM) solution.
- Configure diagnostic settings for each generative AI workflow.
- Use Microsoft Defender XDR, our cloud-native extended detection and response solution, to proactively hunt for threats across generative AI workloads.

Find best practice resources for the implement phase >

Conclusion

Ultimately, securing generative AI workloads is a complex, ever-evolving task. However, by unifying posture management and threat protection under a single cloud-native umbrella, organizations can deliver comprehensive security for their generative AI applications.

Want to learn more about protecting AI against the most pressing threats of today, as well as the emergent threats of tomorrow?

Learn more about Microsoft cloud security solutions.

Resources

Top 5 Generative AI Threats in Runtime

Poisoning Attacks
Azure best practices for generative AI >
Security alerts for AI workloads >
Evaluation and monitoring metrics for generative AI >
Content Credentials in Azure OpenAI >
Abuse monitoring in Defender for Cloud >

Evasion Attacks
Prompt Shields in Azure OpenAI >
Content filtering in Azure OpenAI >
Security alerts for AI workloads >
Abuse monitoring in Defender for Cloud >
How Microsoft approaches AI red teaming >

Functional Extraction and Inversion Attacks
Rate limiting in Azure API Management >
Content filtering in Azure OpenAI >
Abuse monitoring in Defender for Cloud >
Content Credentials in Azure OpenAI >

Prompt Injection Attacks
Python Risk Identification Tool for generative AI (PyRIT) >
Prompt Shields in Azure OpenAI >
Content filtering in Azure OpenAI >
Input validation >
Security alerts for AI workloads >
Abuse monitoring in Defender for Cloud >

Enhancing Generative AI Security in Three Phases

Phase 1: Assess
AI security recommendations in Defender for Cloud >
AI security posture management in Defender for Cloud >
AI Risk Database >

Phase 2: Plan
How to configure content filters in Azure OpenAI >
How to implement logging and monitoring in Azure OpenAI >
How to enable threat protection for AI workloads in Defender for Cloud >
Security alerts for AI workloads >

Phase 3: Implement
Azure Policy >
Threat protection for AI in Defender for Cloud >