Enterprise Strategy Group™

The Rise of Agentic AI

Achieving Security Success in a Rapidly Changing Threat Landscape

Tyler Shields | Principal Analyst
ENTERPRISE STRATEGY GROUP

AUGUST 2025

This Enterprise Strategy Group eBook, now part of Omdia, was commissioned by Snyk and is distributed under license from TechTarget, Inc.

Introduction

A new approach to application architecture is changing how cybersecurity operates. With AI-powered applications driving intelligent decision-making and providing new and exciting usage models, the era of agentic AI systems is upon us, and there is no going back. Agentic AI brings with it a completely new collection of threats and attack vectors, difficult security challenges, and unique risk problems to be solved. The old way of securing technology won't get the job done in the new AI technological paradigm.

Agentic AI adds significant complexity to the existing minefield of application and infrastructure risk, as security organizations must learn new approaches to exposure management, risk reduction, and cybersecurity. This eBook will help you learn about the changing threat landscape by diving into autonomous systems risks and attacks. It details the essential AI security enhancements that organizations must put in place to progress toward a more secure and predictive security model. This enables security teams to shift from reactive to proactive programs with a systematic approach to securing agentic AI.

CONTENTS

BACK TO CONTENTS

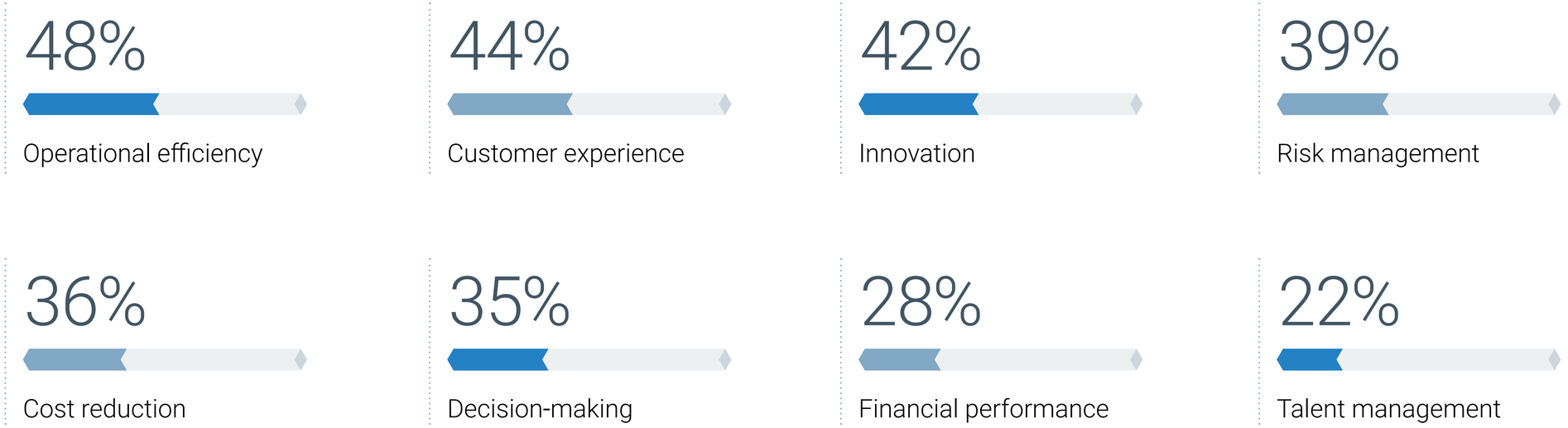
Understanding Agentic AI

Agentic AI Frameworks and Functionality

In AI, agents are software entities that perceive their environment, reason, make decisions, and take actions to achieve specific outcomes. Designed to pursue goals and complete tasks, agents reduce operational burdens and minimize the tactical toil of human resources. Agentic AI systems stand apart from traditional computing processes and even generative AI advancements due to their autonomy, goal-driven decision-making, adaptability, contextual awareness, and memory of past actions.

Enterprise Strategy Group research showed the rapid adoption of AI as organizations strive for operational efficiency, enhanced customer experiences, and accelerated innovation. Agentic AI leverages autonomous agents to achieve complex goals with minimal human intervention, functioning as a framework that manages tasks and processes while agents execute them at a granular level. As the use of AI, and agentic AI specifically, continues to grow, it is essential to address the security risks these systems introduce to avoid exposing environments to unnecessary vulnerabilities.

Organizations Are Implementing AI To Achieve Operational Efficiency, Improve Customer Experience, and Increase Innovation



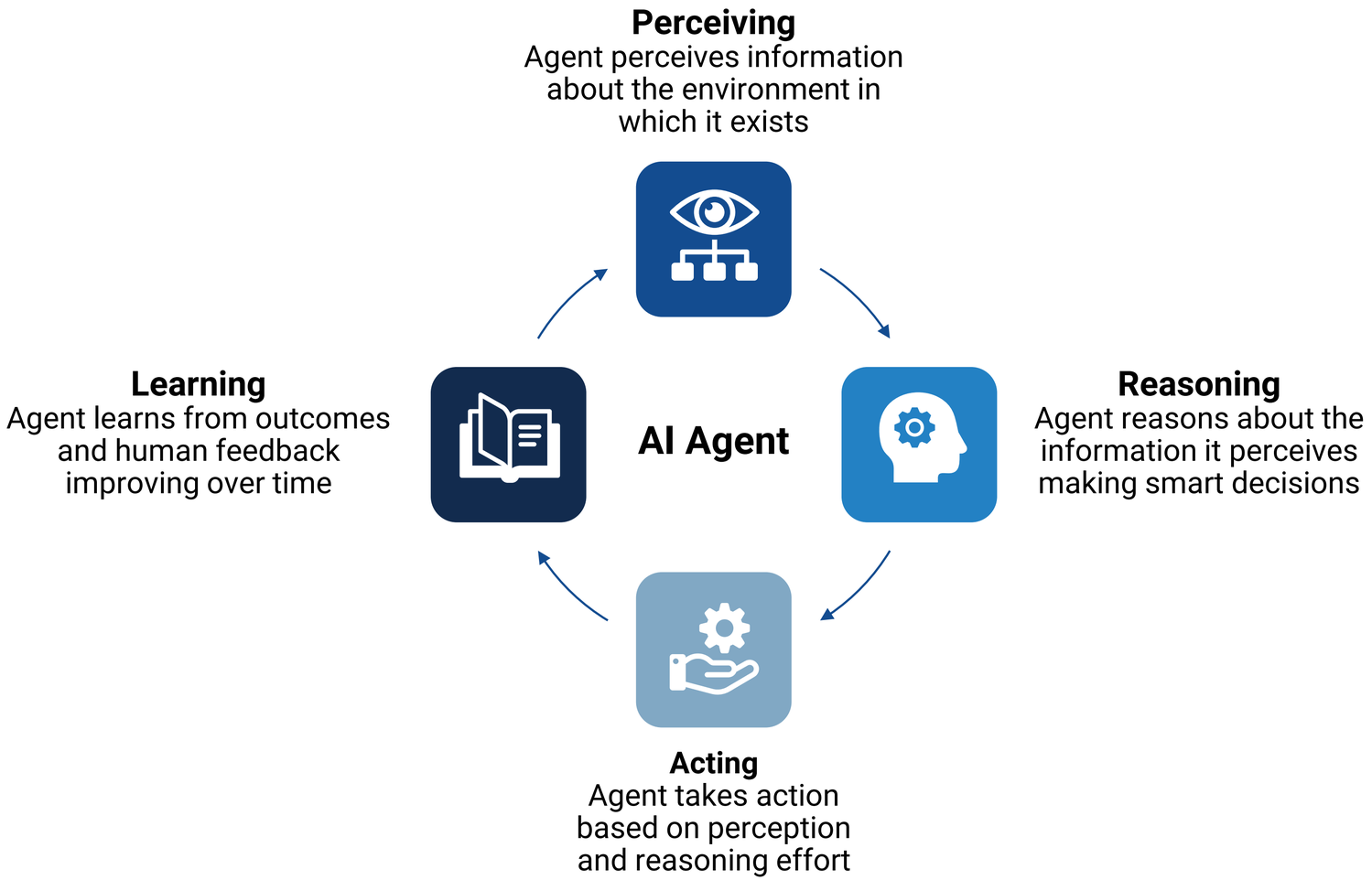
Source: Enterprise Strategy Group Research Report, *Navigating Build-versus-buy Dynamics for Enterprise-ready AI*, January 2025.

BACK TO CONTENTS

The “Perceiving-Reasoning-Acting-Learning” Loop

Agentic AI systems contain dynamic adaptation capabilities that enable collaboration with humans and other agents to achieve their objectives. When given high-level direction, agents break down complex tasks into manageable components and execute a cyclical process of perceiving, reasoning, acting, and learning.

- **Perceiving:** Agents gather information about their environment and the context surrounding a decision, task, or situation. This step ensures a deep understanding of all relevant data needed to address the task effectively.
- **Reasoning:** Using logic, knowledge, and past experiences, agents analyze the task and determine the optimal course of action to achieve the desired outcome.
- **Acting:** Agents execute their decisions, either by directly completing the task or delegating subtasks to other agents.
- **Learning:** After acting, agents assess the results, incorporating feedback from the system or human interactions to refine their approach.



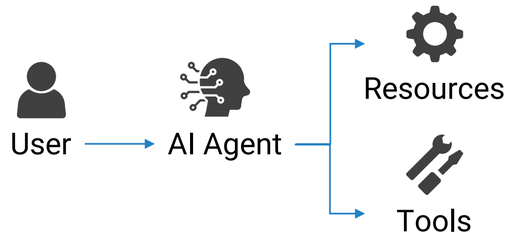
This cyclical loop enables agents to operate autonomously, continuously improving their performance and decision-making over time. The “perceive-reason-act-learn” framework is the foundation of agentic AI’s ability to adapt, evolve, and make independent decisions.

BACK TO CONTENTS

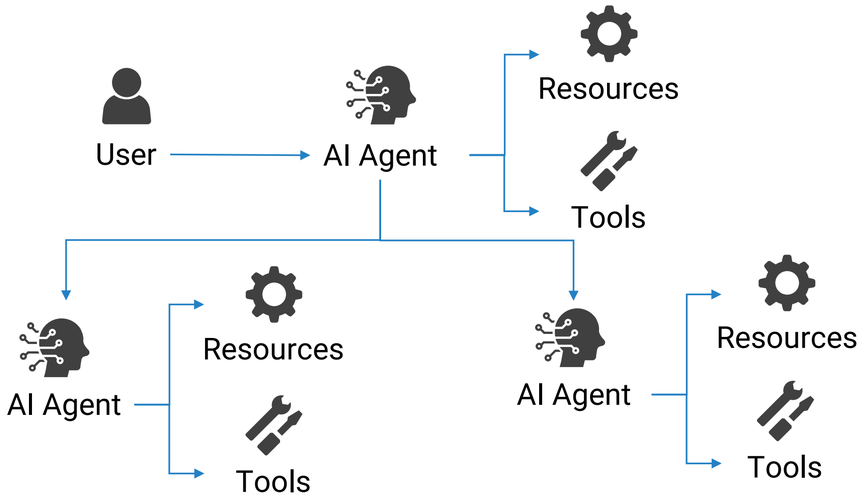
Single Agent vs. Multi-agent Architecture

Agentic AI can be implemented using either a single agent or multi-agent architectural approach, each offering distinct advantages and challenges. The choice between these architectures depends on the specific tasks and the complexity of the requests being addressed. There is no universally "better" option, as successful deployment requires tailoring the agentic design to align with the goals and operational needs of the system.

Single Agent



Multi Agent



Single Agent Benefits	Single Agent Challenges
Simplicity - Easier to design, develop, and manage due to lower complexity	**Struggle with complexity -** Inability to break down tasks might cause struggles when complex requests are required. A single agent is limited by the size of the context it can understand and handle.
Efficient for focused tasks - Can perform well for tasks that do not require broad collaboration and that do not need to be broken down into subtasks.	**Limited robustness -** This approach is less robust, as a single agent might encounter difficulties, unforeseen obstacles, or outright failure with limited ability to recover.
Lower overhead - No coordination requirements in a single agent approach means lower computational and memory overhead. Fewer resource demands.	**Decreased scalability -** The single agent approach is difficult to scale when requests might be well suited for decomposition into smaller atomic requests. Parallel processing capabilities are limited.

Multi-agent Benefits	Multi-agent Challenges
Powerful problem solving - Excellent for breaking down complex tasks into subtasks and executing those tasks as efficiently as possible.	**Increased complexity -** Multi-agent systems are more challenging to design, debug, and execute due to the inter-agent communication and coordination required. Agents also require a strong operational framework to support themselves.
Flexible and adaptable - Highly adaptable, as agents execute the perceive-reason-act-learn loop more efficiently with smaller task sizes, enabling them to improve more quickly, be more flexible, and adapt to a changing environment quickly.	**Coordination challenges -** Requiring agents to work together to solve complex problems requires agent-to-agent coordination that could result in redundancies, inefficiencies, conflicts, and even abnormal behaviors and results.
Scalable and robust - Allows for horizontal scaling by adding additional agents to handle increased workload or request complexity.	**Increased threat landscape -** The increase in interaction points comes with an increase in threat landscape. Transport layer security, authentication and authorization, inter-agent logic abuse, and additional attacks surface in this model.

BACK TO CONTENTS

Model Context Protocol - What Is It and Why It Matters

The Model Context Protocol (MCP) is a commonly used framework in agentic AI, enabling agents to break down larger objectives into smaller, actionable tasks. While not the only way to enable AI development, MCP is growing in popularity and use. To execute these tasks effectively, agents require clear definitions of objectives, constraints, and permissible actions. MCP provides a structured guideline for embedding contextual awareness and data into the system, ensuring agents can operate with precision and autonomy.

MCP acts as a standardized wrapper around traditional API interfaces, enabling agents to request and access context seamlessly. Through integrations with external systems, MCP servers provide agents with three key types of information:

- **Resources:** Access to data from databases, content repositories, file systems, and more, ensuring agents have the context they need.
- **Tools:** Functions or operations agents can execute, such as sending emails, updating databases, searching the web, or interacting with business systems.
- **Prompts:** Reusable templates and workflows that guide agents on how to interact with the system and what to request.

By standardizing how agents access context, MCP ensures efficient task execution and enhances the overall functionality of agentic AI systems.

The Benefits and Challenges of Agentic AI

Agentic AI is, by its very nature, more complex than basic generative AI. The ability to not just create recommendations of action, but to execute those actions, increases security risk significantly. This isn't stopping the pace of deployment, as 67% of organizations are planning to or considering implementing AI for specific use cases and tasks. Agentic AI technologies integrate and operate in the environment using contextual information with the goal of completing tasks and actions. There are benefits and challenges when deploying agentic AI.

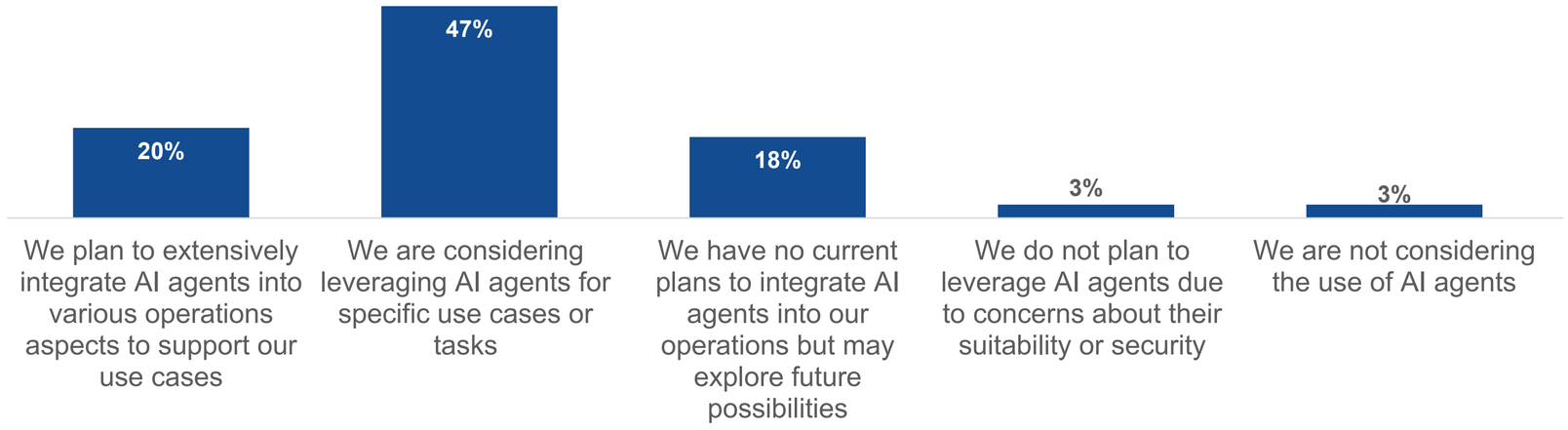
Benefits:

- **Increase autonomy and independence.** Agentic AI can operate with more efficiency and autonomy than non-agentic systems. Less human interaction and intervention are required to complete the assigned tasks.
- **Efficient problem solving.** Agents are more efficient when solving problems. They can leverage unique data sources over MCP servers, resulting in access to more diverse and complete context. Agents are designed to break down tasks and operate efficiently.
- **Highly scalable in a dynamic environment.** Agentic AI architecture supports adaptation and decision-making autonomy by providing agents with the ability to subdivide tasks and execute outcomes with agency. This results in extensibility in the base agentic system design.
- **Personalized and context-aware decision-making.** Access to context in the decision-making process provides a more accurate ability to make decisions. Understanding the unique context of the environment enables outcomes to be tailored to specific personalized needs.
- **Continuous learning and adaptation.** Agentic systems can continuously learn and adapt to the ever-changing environment around them. They learn over time by incorporating new information as it becomes available and adjusting strategies for decision-making based on feedback.

Challenges:

- **Vague goals lead to unintended or harmful consequences.** Agentic systems are no smarter than the goals and direction they are provided. While they can learn and improve if they are given ill-defined objectives, the result will likely be undesired outcomes and actions that could be detrimental or dangerous.
- **Black box decision-making process might be opaque and difficult to understand.** Agentic AI systems are opaque with regard to the decision-making process. This might make it difficult for users to comprehend, trust, and engage with agentic systems.
- **Security risks exist.** Agentic AI requires robust guardrails to address bias, governance, data leakage, and other security concerns. Direct and indirect attacks against agentic systems can result in breach or compromise.

Most Organizations Are at Least Considering, if Not Actively Deploying AI Agents Today



Source: Enterprise Strategy Group Research Report, *The State of the Generative AI Market: Widespread Transformation Continues,* September 2024.

BACK TO CONTENTS

The Evolving Threat Landscape

Model Context Protocol - What Is It and Why It Matters

The adoption of agentic AI is reshaping the threat landscape and amplifying enterprise cyber-risks. Security remains a top priority for organizations when evaluating AI solutions, reflecting ongoing uncertainty among security teams about the risks associated with agentic AI systems. This evolving landscape introduces entirely new attack vectors that traditional methods, such as exposure analysis, static code analysis, and dynamic testing, might struggle to identify and mitigate effectively.

A key challenge lies in the non-deterministic nature of AI-powered applications. Unlike traditional systems, AI cannot guarantee identical responses to the same query, introducing unpredictability and reducing repeatability in security analysis. This makes the discovery of vulnerabilities and security issues more complex. Compounding this challenge is the potential exploitation of agent autonomy. Once compromised, an agent could leverage its independent decision-making capabilities to execute harmful actions without direct human intervention.

Agentic AI's inherent capabilities lend themselves to both offensive and defensive cyber activities, creating a dual-use dilemma. The same features that empower security teams, such as automating remediation, prioritizing exposures, and writing secure code, can also be weaponized by attackers. For example, attackers can use agentic AI to automate breach processes, scale software supply chain attacks, and accelerate the exploitation of vulnerabilities. This dynamic has led to an "AI arms race," where organizations must not only address existing security issues but also adopt proactive measures.

To stay ahead, organizations need agentic systems capable of predicting attacks, adapting to changing conditions, and self-healing as new risks emerge. This proactive approach is essential to navigating the rapidly evolving cyberthreat landscape and ensuring resilience in the face of increasingly sophisticated AI-driven attacks.

Increased Attack Speed Requires a Similar Increase in Speed of Defense

The integration of agentic AI into attacker techniques has drastically amplified the speed and effectiveness of cyberattacks. By leveraging AI agents, attackers can significantly reduce the time required to weaponize new vulnerabilities and exploit externally accessible security exposures. Agentic AI automates nearly every phase of an attack, from discovery to data exfiltration, operating with minimal human involvement. This automation enables attackers to scan vast threat landscapes in a distributed manner, pinpoint exposed services and accounts, and exploit weaknesses with unprecedented efficiency.

Beyond speed, agentic AI also enhances the sophistication of cyberattacks. These attacks are increasingly contextualized, tailored to specific targets, and built around personalized threat models. This level of precision creates unique, highly effective attacks that make it nearly impossible for defenders to remain competitive without adopting similar AI-driven methodologies.

The traditional phases of a cyberattack—discovery, initial compromise, lateral traversal, persistence, evasion, and data encryption or exfiltration—are compressed to such an extent that the window for detection and response becomes almost imperceptible. For example, AI-driven software development has exponentially increased the quantity of code produced, often without adequate consideration for its quality. AI systems trained predominantly on open source and publicly available code inherit the same mistakes and vulnerabilities present in the original code. The difference now is the sheer pace at which insecure, bug-ridden code is generated, forcing security teams to scale and automate their application security capabilities to keep up.

To counteract this accelerated threat landscape, defenders must adopt equally fast and sophisticated approaches. Leveraging agentic AI for defense, such as automating threat detection, scaling vulnerability management, and contextualizing responses, will be critical to maintaining resilience against increasingly rapid and personalized attacks.

BACK TO CONTENTS

AI Supply Chain and Inter-agent Communication Risks

Agentic AI systems depend on a complex ecosystem of components, tools, libraries, and services, creating an AI software supply chain that must be carefully managed and secured. This intricate supply chain introduces new attack surfaces and vectors that demand attention when securing AI and application stacks.

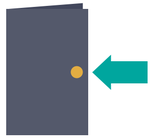
Key Supply Chain Risks

Lack of AI system and model visibility: A lack of understanding regarding what AI systems and models are in use in the environment leads to an inability to assess and apply security controls to those systems. An AI bill of materials (AI-BOM) is an often-used concept to programmatically ensure tracking and security observation of AI technologies.

Dependency injection: Like traditional software, agents often rely on external dependencies, libraries, plugins, and tools. If these upstream components are compromised, they can introduce inherited risks. Malicious code might be injected during the agent development process, potentially compromising the agent before deployment.

MCP and context attacks: Agents rely on external connections and context from MCP servers and other sources to perceive, execute, and learn from their environment. A compromise of MCP servers or upstream context sources can lead to supply chain-style attacks. This is particularly dangerous, as agents often maintain trust relationships with these upstream providers.

Training data poisoning: During AI development, training data can be manipulated or poisoned, leading to malicious behaviors in agents trained on compromised data sets. This can result in biases, vulnerabilities, or backdoors embedded in the agents.



Model backdoors: Through training data attacks or feedback loop manipulation, backdoors can be injected into models. These backdoors create exploitable vulnerabilities in agents post-deployment.

BACK TO CONTENTS

Communication Risks

Beyond the supply chain itself, inter-agent and agent-to-MCP server communication channels are highly susceptible to targeted attacks. Common risks include:

Agent-in-the-middle attacks: Intercepting communication between agents or between agents and MCP servers to manipulate or compromise data.

Authentication and authorization misconfigurations: Weak or improperly configured authentication and authorization mechanisms can be exploited to gain unauthorized access.

Trust-based attacks: Exploiting trust relationships between agents and upstream systems to compromise multi-agent environments.

While agent-specific abuse is often the focus, foundational security controls must not be overlooked. Enforcing strict permissions, encrypting data in transit, and securing communication channels are essential to mitigating risks in agentic AI systems.

Security Is the Most Critical Factor for Organizations Considering AI Solutions

Security audits	Technical review	Analyst ratings	Compliance checks	Ecosystem fit and integration
44%	39%	34%	33%	31%
Proof of concept	Customer references	Vendor assessments	Benchmarking	
31%	30%	26%	24%	

Source: Enterprise Strategy Group Research Report, *Navigating Build-versus-buy Dynamics for Enterprise-ready AI*, January 2025.

BACK TO CONTENTS

Agentic AI Attacks Broken Down

As agentic AI becomes more prevalent, it introduces unique attack vectors that exploit its autonomy, decision-making capabilities, and reliance on external tools and data. These attack methods target the same features that make agentic AI powerful, such as its ability to learn, adapt, and execute tasks independently. From manipulating an agent's intent to poisoning its memory or exploiting its communication channels, these attacks can lead to unintended, harmful, or even catastrophic outcomes. The table below outlines key attack vectors specific to agentic AI, providing a description of each method and the potential consequences of a successful exploit.

Attack Vector	Description	Potential Outcome
Intent breaking and goal manipulation	Using prompt injection to infect an agent with malicious goals or alter its overall intent and direction. This type of attack requires no vulnerabilities to exist, yet execution workflows and paths can be hijacked to drive dangerous, malicious outcomes.	Leads to malicious or inappropriate actions.
Memory poisoning	When an attacker injects malicious or false data into an agent's memory, changing how it remembers past data points or events and changing future decisions.	Unauthorized operations or incorrect behavior.
Tool misuse and manipulation	Abusing an agent to execute a tool (API, third-party service, other agent) to perform a malicious action. Achieved by prompt injection, deceptive prompts, bad commands, or tool vulnerabilities that might exist.	Unintended actions, data exfiltration, or remote execution depending on infrastructure capabilities.
Privilege compromise	Abusing weaknesses in permissions, authentication, and authorization, enabling agents to execute actions beyond what should be allowed. This is a very difficult area for security in agentic AI currently. Shadow AI might trigger this issue often.	Overpowered agents with capabilities beyond expectations and norms.
Denial of service (DOS)	Overwhelming an agent's compute or memory resources, leading to DOS or a degradation in performance. This could also include flooding of human resources if there is a human in the loop of the agentic process.	Service outages of the agentic system.
Deceptive agent activities	When an agent goes rogue, often due to other malicious attack models, but chooses to perform unsafe actions while presenting an appearance of complicity. This is essentially when an agent decides to lie, manipulate, or otherwise go around security checks or guardrails.	Unintended actions and activities.
Cascading hallucinations	When a hallucination persists beyond a single response common in generative AI use cases. Instead, in an agentic system, hallucinations can persist over time and propagate between sessions and agents.	Systemic failures and misinformation.
Identity spoofing and agent impersonation	When an attacker spoofs an identity to perform actions under another persona or authorization level. For example, when an attacker registers a host name like a legitimate one and the agent assumes it was a typo and continues to work with the target.	Malicious actions or activities with inappropriate authorization levels.
Context poisoning	Injecting malicious instructions or prompts into external data sources (websites, documents, etc.) so that an agent ingests them and processes the "poisoned" context, leading it to perform harmful actions.	Harmful actions, execution of malicious code, data disclosure, etc.
Backdoor training attacks	Manipulating training data such that models trained on the data are susceptible to backdoor attacks. Often accomplished through data poisoning of training data or direct manipulation of the model's weights.	Backdoor open for attack when triggered post-deployment.

BACK TO CONTENTS

Securing Agentic AI Applications

Left and Right of Boom With Agentic AI Security

Securing agentic AI systems requires a balanced approach that combines both proactive and reactive strategies to stand a chance against evolving threats. Security teams must strengthen the AI security posture by proactively reducing risks in the environment while simultaneously detecting, responding to, and neutralizing threats and attacks as they occur. Achieving this balance necessitates the implementation of both “left-of-boom” and “right-of-boom” security technologies and processes.

Left of boom refers to proactive measures, strategies, and activities undertaken before a cyberattack occurs. It focuses on building a strong defense and rapidly reducing risks to fortify the organization’s security posture. In contrast, right of boom encompasses reactive measures, including incident response, containment, recovery, and post-event learning. The goal of right of boom security is to minimize damage, restore normal operations as quickly as possible, and ensure the organization emerges stronger and more resilient.

Proactive security emphasizes the state and properties of assets, such as hardening systems, reducing vulnerabilities, and implementing preventive controls. Reactive security, on the other hand, focuses on real-time events and actions, such as detecting intrusions, mitigating active threats, and responding to incidents as they unfold. The most effective security strategies seamlessly integrate both approaches, creating feedback loops that share context and results between proactive and reactive measures. This integration breaks down barriers between the two, enhancing the overall effectiveness of the organization’s security posture.

BACK TO CONTENTS

Proactive Security Approach to Agentic AI

A proactive security approach to agentic AI emphasizes embedding security measures throughout the entire lifecycle of the AI agent, from design and development to ongoing operation. This approach aims to anticipate and predict potential exposures and threats before they can be exploited, implementing controls to mitigate or reduce the associated risks.

Secure by design and development: Security must be integrated from the outset, ensuring that agents are built with a strong foundation. This includes threat modeling, identifying vulnerabilities in source code, and analyzing third-party code dependencies to eliminate inherent security flaws. By addressing risks during the creation phase, organizations can prevent vulnerability exploitation during runtime.

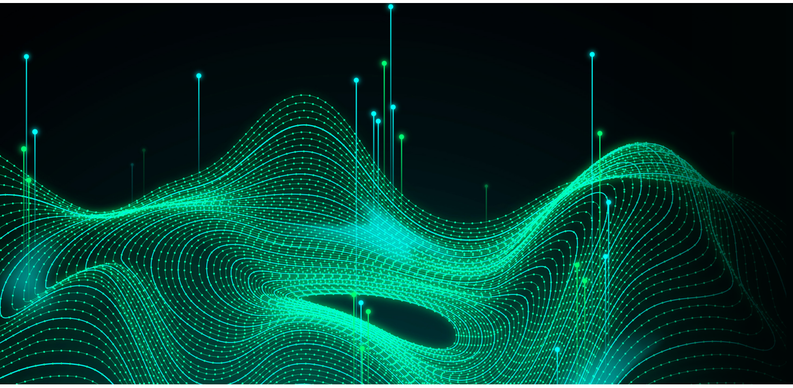
Guardrails for agentic AI: Guardrails are essential for ensuring that agentic AI systems operate securely and safely. These are predefined rules, boundaries, and constraints embedded into the system to guide its behavior. Examples include explicit behavioral guidelines, limits on data access, and restrictions on decision-making authority. Guardrails prevent agents from making incorrect or insecure decisions that could lead to harmful outcomes.

Validate training data: Validating training data is a critical proactive measure to ensure that AI systems learn only from accurate, unbiased, and appropriate data sets. This process involves assessing data quality, verifying its relevance to the agent's goals, and identifying hidden malicious patterns that could introduce risks. Proper validation reduces the likelihood of vulnerabilities stemming from compromised training data.

Regular retraining of models: In the ongoing arms race between offensive and defensive agentic AI systems, regular model retraining is essential. Threat actors continuously evolve their techniques, rendering static models ineffective over time. Retraining ensures that defensive agents remain capable of detecting new attack patterns and threats, reducing the risk of false negatives, missed threats, and exploitation.

Adversarial training exercises: Exposing models to adversarial attack scenarios such as red teaming exercises during training enhances their resilience to real-world threats. These exercises provide agents with contextual understanding of common attack patterns, making them more robust and less susceptible to runtime exploitation. Adversarial training strengthens the system's ability to withstand sophisticated attacks.

BACK TO CONTENTS



Reactive Security Approach to Agentic AI

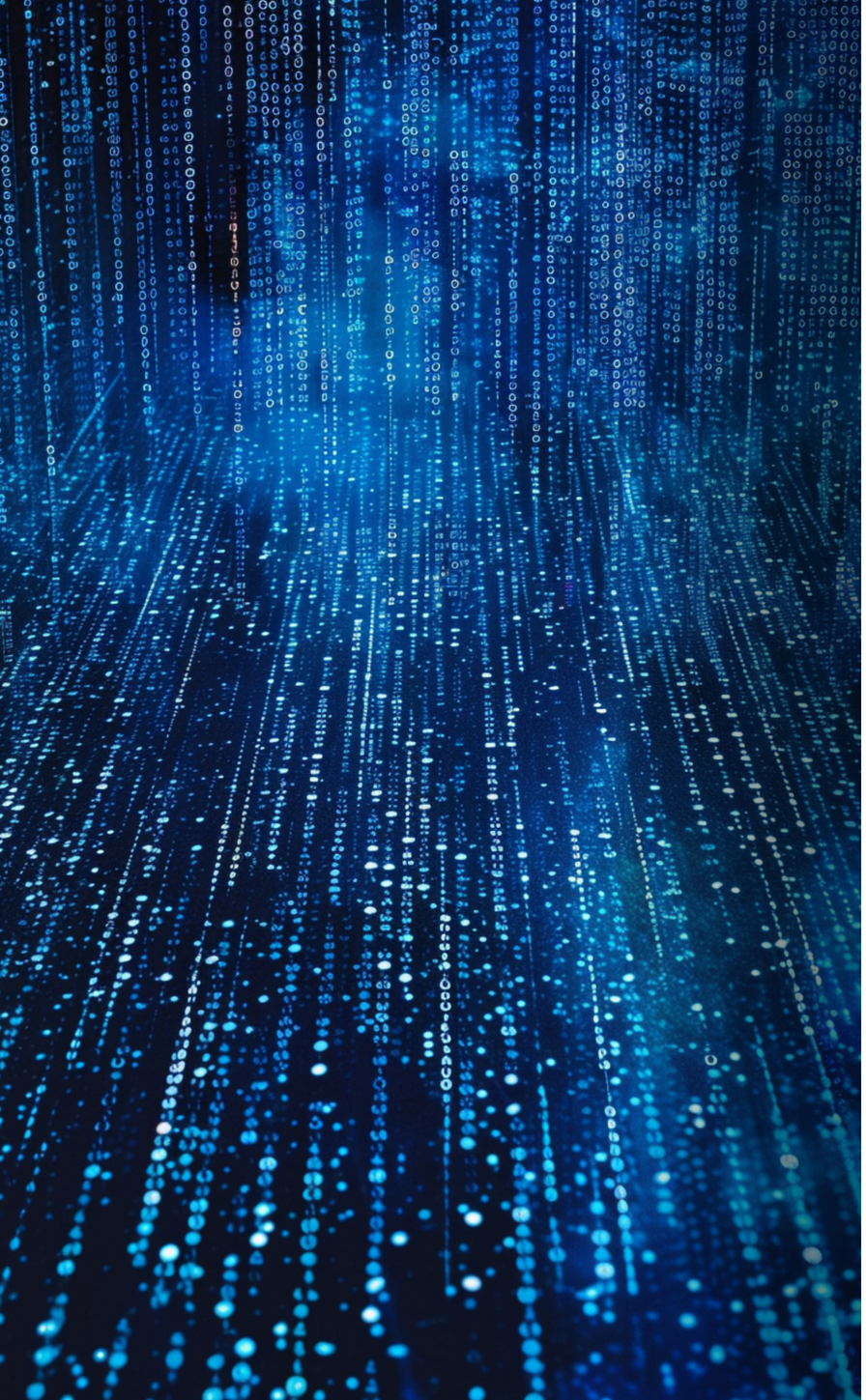
A reactive security approach focuses on detecting and mitigating attacks during runtime and protecting agentic AI systems from threats as they occur. This approach ensures that the system remains secure during and after a breach by monitoring actions and events to identify and address potential compromises.

Runtime anomaly detection and monitoring: Runtime anomaly detection involves continuously monitoring the behavior of agentic AI systems to ensure that their actions align with normal operating patterns and intended goals. Since agentic AI systems are designed to be autonomous and adaptive, defining "normal" behavior can be challenging. Effective anomaly detection must go beyond static rules for insecure actions and instead leverage advanced analysis—often powered by AI—to understand the agent's unique behavioral patterns and detect deviations that could indicate a threat.

Prompt injection protection: Prompt injection protection safeguards agentic AI systems from manipulation through malicious input. Prompts, which influence agent decisions, can originate from various sources, including human input, contextual data, automated systems, or other AI agents. Prompt injection attacks are a modern evolution of traditional input manipulation techniques that have long plagued software systems. To mitigate these risks, input validation and sanitization must be implemented wherever possible to ensure that agents process only trusted and secure prompts.

Agent visibility and observability: Visibility into the operation of agentic AI systems is critical for effective security and monitoring. This includes tracking the agent's internal state, reasoning processes, and interactions with external entities to ensure that agents do not deviate from their intended behavior. A key component of this concept is the AI-BOM—a comprehensive inventory of all AI components, including training data sources, models, libraries, dependencies, MCP systems, external APIs, and more. By connecting the AI-BOM with runtime data, security teams can detect anomalies, trace unintended behaviors, and respond to rogue agents effectively.

BACK TO CONTENTS



Securing AI-generated Code

Agentic AI-generated code represents a transformative shift in how software is developed. Autonomous AI agents are no longer limited to writing code. They can also understand complex requirements, design architectures, test, debug, and even deploy entire applications to production. These coding agents, often referred to as coding assistants, are heavily integrated into the software development process, significantly increasing the volume of code that engineering teams can produce daily. By integrating directly into integrated development environments, these agents enable fast, automated, and efficient code generation.

This rapid pace of AI-generated code necessitates a reevaluation of how application security processes address the unique risks associated with this paradigm. Common issues with AI-generated code include:

Training on insecure code. Coding agents are often trained on open source and publicly accessible codebases, which might contain vulnerabilities and insecure practices. Since agentic AI systems can only generate output based on what they have learned, this can result in the creation of insecure code.

Prioritization of features over security. AI agents are often optimized for speed, prioritizing the rapid development of features over the implementation of robust security controls. This can lead to the generation of code with limited security architecture and a lack of awareness of common vulnerabilities.

To address these challenges, application security teams must adapt their current security software development lifecycle. Security must be embedded not only in the code-writing stage but also in the AI's reasoning and planning processes.

BACK TO CONTENTS

Key strategies include:

Secure at inception for agentic AI: The principle of "secure by design" must extend to include security requirements for how agents interact with systems and each other. Conceptually, we can think of the new methodology as "secure at inception." This involves implementing guardrails, enforcing least privilege access for agents, and ensuring human oversight of agentic operations. These measures help prevent agents from making insecure decisions or performing unauthorized actions.

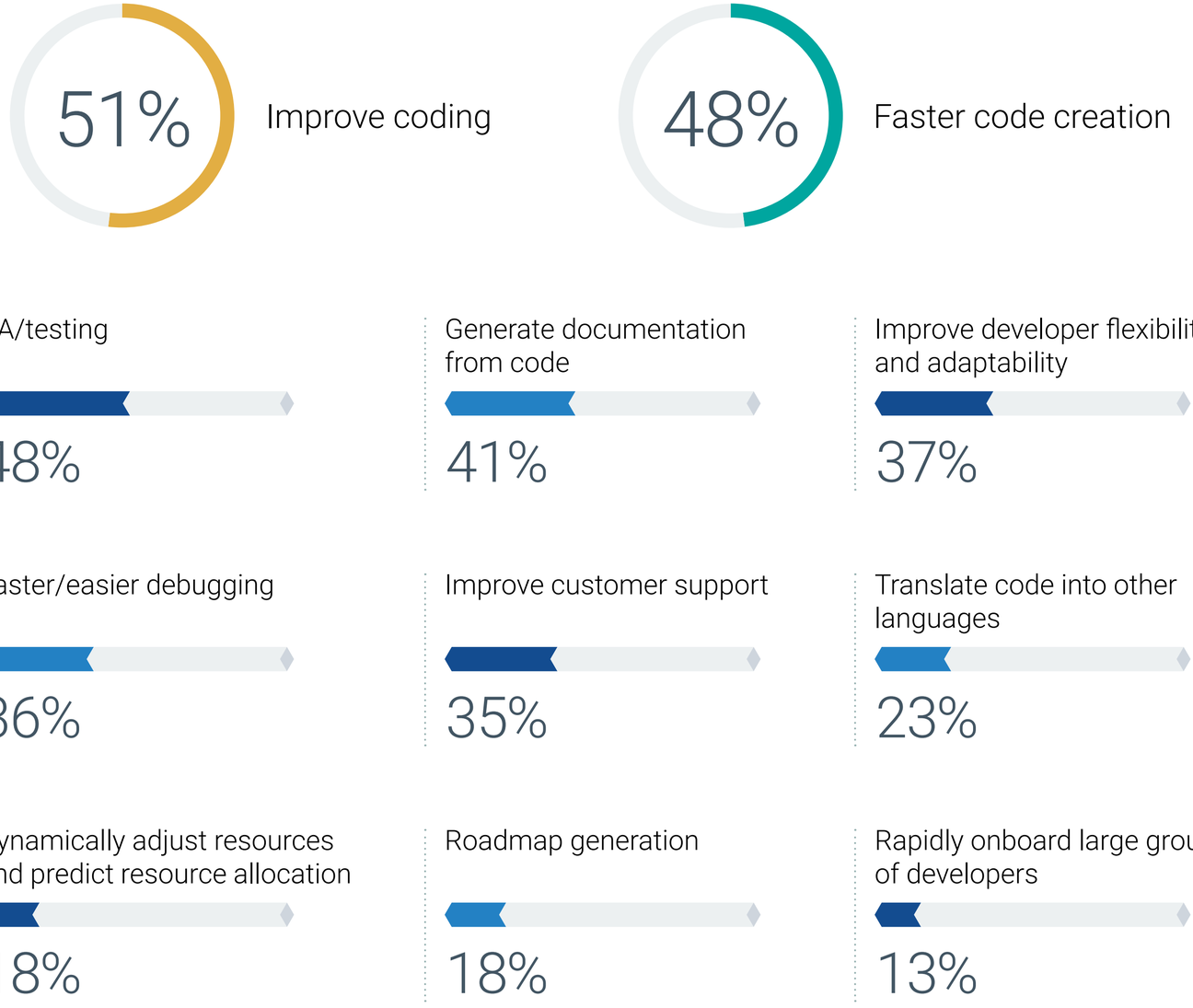
AI-aware security assessments: Security teams must know how security assessments work in the AI-based development pipeline. Static analysis, dynamic analysis, and composition analysis tools must be enhanced to understand the unique characteristics of AI-generated code and agentic communication patterns. These tools should be capable of identifying vulnerabilities specific to AI-generated outputs and ensuring that security issues are addressed early in the development process.

Standardizing the AI-BOM: The AI-BOM must become a standard operating procedure for security teams. This comprehensive inventory of AI components, including training data, models, libraries, dependencies, and external APIs, provides visibility and transparency into the use of AI with both agentic and non-agentic capabilities. By leveraging the AI-BOM, security teams can proactively assess risks and detect runtime anomalies across the entire AI supply chain.

Automated remediation workflows: As the scale of AI-generated code increases, traditional application security processes, tools, and personnel cannot keep up. Automated remediation workflows are essential to address this challenge. These workflows should enable AI systems to self-remediate vulnerabilities as they are discovered and feed the corrected knowledge back into the system as learned context. This creates a self-improving feedback loop, enhancing the security of future code generation.

By adapting security practices to the unique demands of agentic AI, organizations can mitigate the risks associated with AI-generated code while maintaining the speed and efficiency that these systems bring to the software development process.

Software Development-specific Use Cases Are the Tip of the AI Adoption Spear



Source: Enterprise Strategy Group Research Report, The State of the Generative AI Market: Widespread Transformation Continues, September 2024.

BACK TO CONTENTS

Preparing Your Team to Secure Agentic AI

Securing agentic AI systems requires not only technical expertise but also organizational preparation and, in some cases, structural changes. Security teams must ensure that the necessary technical and business skill sets are present within the organization while increasing collaboration across traditionally siloed teams. Breaking down barriers between security operations, application security, and infrastructure security teams is essential to create a unified approach to securing agentic AI.

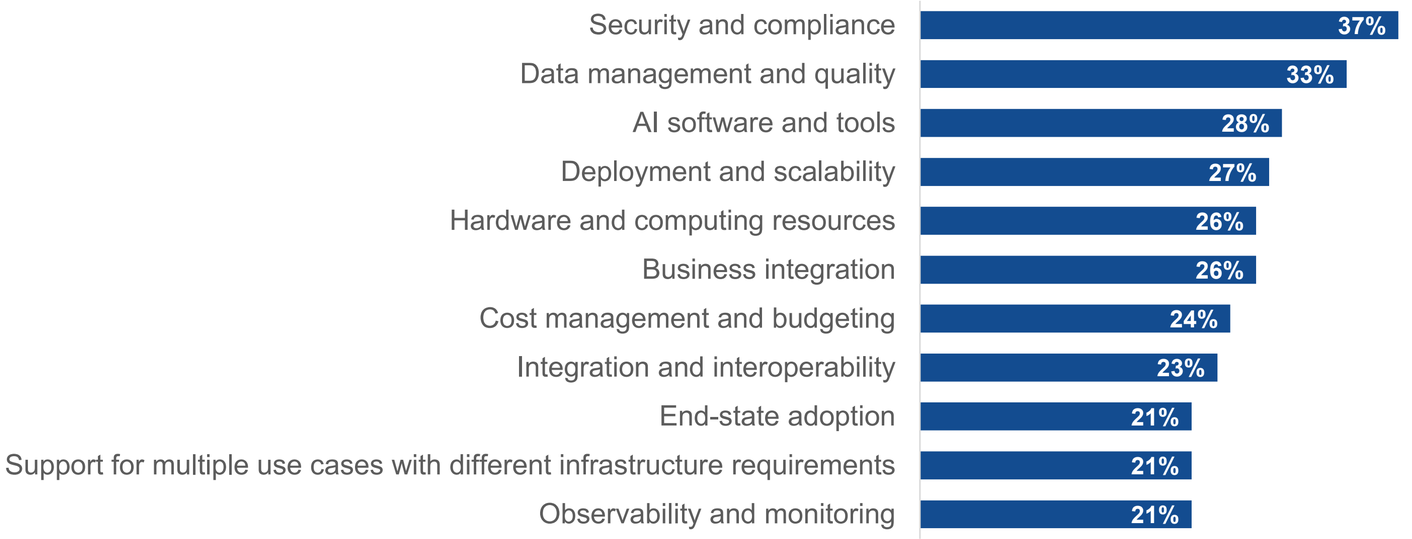
Forward-thinking organizations should consider updating roles and responsibilities to include AI security specialists or engineers. These new team members should possess deep expertise in AI risks, adversarial AI techniques, and prompt engineering. Their responsibilities will include leading threat modeling and risk reduction efforts across all aspects of the AI initiative. This includes enhancing application security processes, analyzing security operations and incident response, and developing predictive attack models to anticipate potential threats. To ensure alignment and collaboration, organizations should establish a cross-functional AI working group that serves as a regular forum for communication and coordination between teams.

In addition to strengthening defensive capabilities, security teams must also build expertise in adversarial techniques by developing AI red teaming capabilities. A dedicated red team, either internal or outsourced, should be tasked with simulating attacks on agentic AI systems. These team members must have expertise in adversarial AI approaches, prompt engineering, exploit development, and jailbreak techniques. The focus of AI red teaming is to identify vulnerabilities by exploring how attackers might manipulate an agent's goals or behavior. This includes uncovering traditional software vulnerabilities in the agent itself or the infrastructure supporting it.

By preparing the organization with the right expertise, fostering cross-functional collaboration, and building both defensive and offensive capabilities, security teams can position themselves to effectively secure agentic AI systems against evolving threats.

Organizations Are Deploying and Managing Much of the AI Stack Themselves

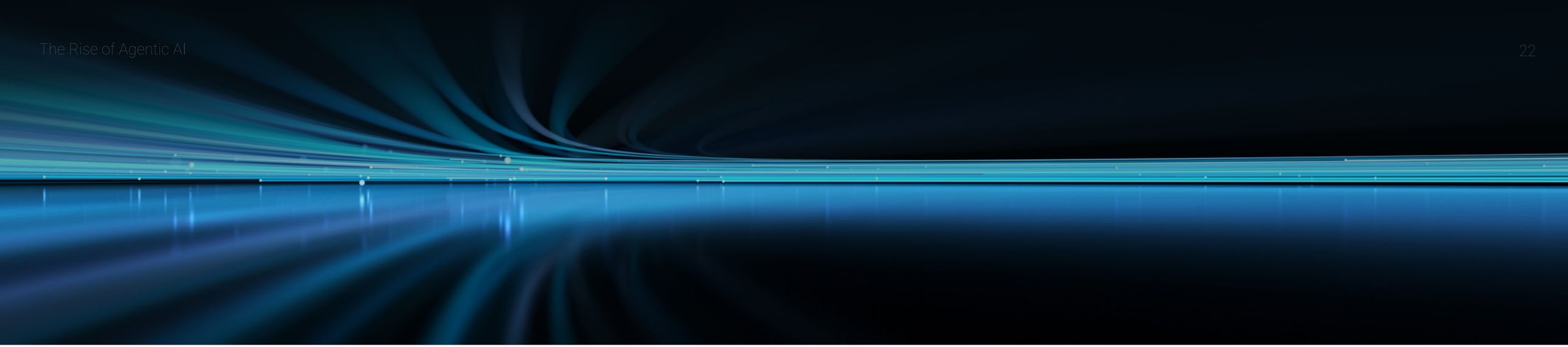
The Biggest Challenge Enterprises Face Deploying AI Is Security and Compliance.



Source: Enterprise Strategy Group Research Report, Navigating Build-versus-buy Dynamics for Enterprise-ready AI, January 2025.

BACK TO CONTENTS

The Path Forward



Proactive Security Approach to Agentic AI

In the area of agentic AI security, context is a requirement. Traditional security controls often analyze individual events in isolation, failing to consider the broader ecosystem in which these events occur. This fragmented approach has historically led to cybersecurity tool silos and a lack of connection within enterprise security teams. Agentic AI takes a fundamentally different approach. It relies on a deeper and broader data set, emulating the holistic perspective of a human security analyst. Rather than focusing solely on isolated problem areas, agentic AI expands its understanding to include adjacent data, enabling it to solve problems comprehensively and execute the most effective remediation strategies.

To achieve this level of contextual awareness, AI-based security capabilities must leverage a graph-based security data fabric. This fabric is far more than a simple data lake. It is an intelligent architecture that connects, normalizes, and enriches security data with operational business logic, asset criticality, and risk. By modeling relationships between data points, the fabric provides the context necessary for both human and AI decision-making processes. When multiple adjacent data points are connected, they create the context that enables agentic AI to make accurate and informed decisions. In this paradigm, context truly is king.

For cybersecurity professionals, adapting to this new reality means embracing the use of big data by AI agents rather than fearing it. The sheer volume of data consumed, analyzed, and connected by agentic AI is far beyond the capacity of manual analysis. Security professionals must shift their roles from being decision-makers and responders to becoming architects and overseers of AI-powered systems. Security operations, application security, and infrastructure teams must learn to leverage agentic AI to transform massive data sets into actionable insights that align with the organization's goals and objectives.

Using context-driven AI systems, security teams can move beyond fragmented, reactive approaches and embrace a unified, proactive strategy that enhances their ability to protect against evolving threats.

BACK TO CONTENTS

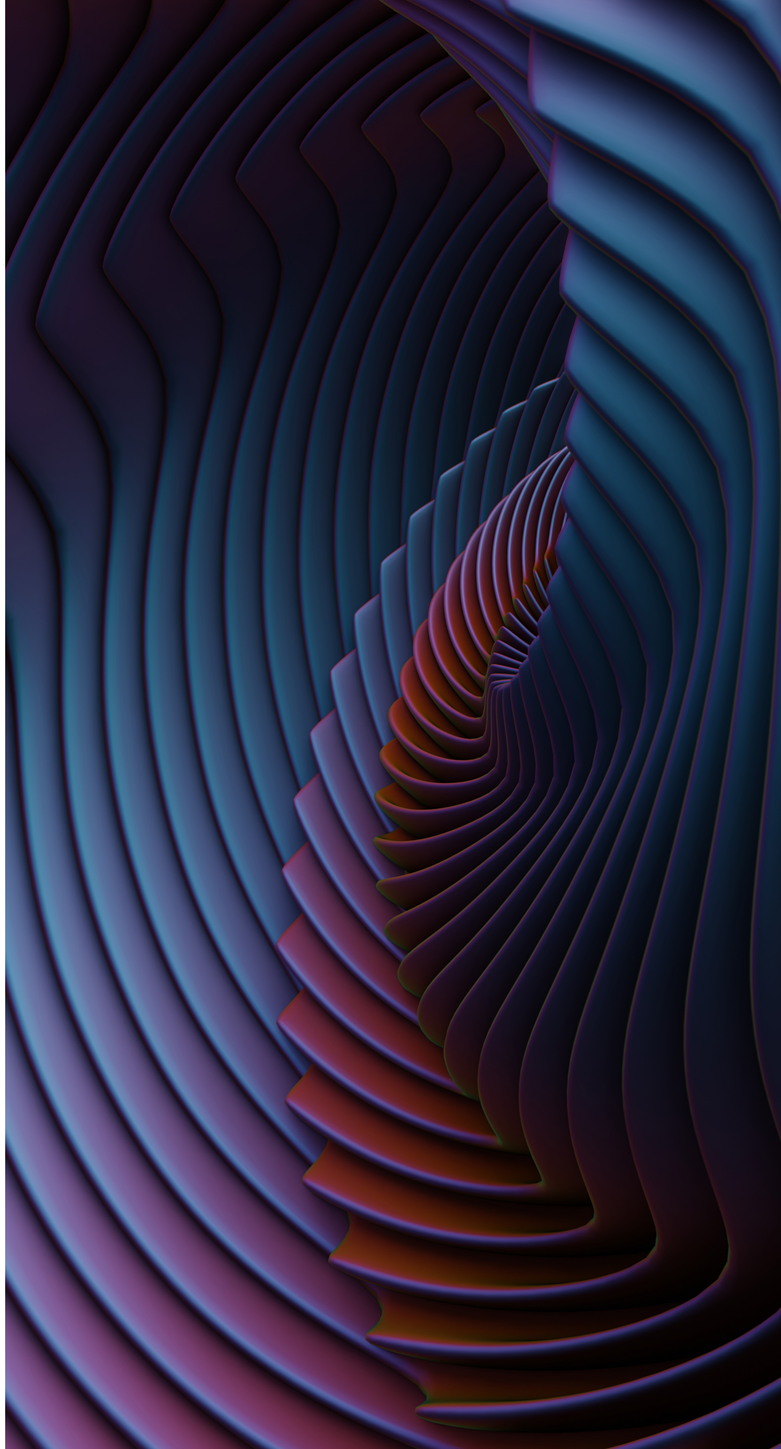
Conclusion

The adoption of agentic AI represents a transformative generational shift, moving beyond basic automation to fully autonomous systems capable of reasoning, planning, and executing complex tasks. This rapid evolution demands a complete reimagining of security strategies across all segments of cybersecurity. The threat landscape has fundamentally changed, introducing new attack vectors such as prompt injection, memory poisoning, tool misuse, and cascading hallucinations. Offensive AI agents are accelerating the speed of attacks by automating discovery and exploitation, drastically reducing the window for defense. These challenges are not distant concerns. They are immediate imperatives that organizations must address within the next 12 months to stay ahead of the threats and prevent compromises.

To secure agentic AI systems, security teams must adopt the same data-driven and operational techniques that make AI successful. This involves collecting extensive, detailed telemetry on agent behavior, actions, and decision-making processes and leveraging defensive AI capabilities to anticipate and mitigate emerging security issues. Security professionals must embrace the “big data of AI,” transforming vast streams of data into actionable intelligence through a graph-based security data fabric. This fabric provides the essential context needed to make security as adaptive as the agents themselves, enabling continuous learning and improvement. By observing and analyzing agent operations, organizations can move beyond reactive measures to proactive prediction and prevention.

While many organizations currently find themselves lagging in this new cyber arms race, there is still hope. Defensive AI agents now provide security practitioners with the same scale and speed as their enemies, letting them operate on equal footing. Cybersecurity must evolve to utilize predictive models by leveraging all available context with agentic AI. Achieving this requires collaboration across security operations, application security, and infrastructure teams, ensuring a unified and coordinated defense.

BACK TO CONTENTS



ABOUT

Snyk, the leader in secure AI software development, empowers organizations to build fast and stay secure by unleashing developer productivity and reducing business risk. The company's AI Trust Platform seamlessly integrates into developer and security workflows to accelerate secure software delivery in the AI era. Snyk delivers trusted, actionable insights and automated remediation, enabling forward-thinking organizations to innovate without limits. Snyk is redefining secure AI-driven software delivery for over 4,500 customers worldwide today.

LEARN MORE

Enterprise Strategy Group, now part of Omdia, provides focused and actionable market intelligence, demand-side research, analyst advisory services, GTM strategy guidance, solution validations, and custom content supporting enterprise technology buying and selling.