AI for Security and Security for AI: Navigating Opportunities and Challenges

November, 2025

SANS

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Introduction

The emergence of AI as a transformative force is leading organizations to rethink security. While AI technologies can augment human expertise and increase the efficiency of security operations, they also introduce risks ranging from lowering technical barriers for threat actors to producing inaccurate outputs.

As AI adoption accelerates, adapting security and compliance strategies to the opportunities and challenges it presents is paramount.

- According to IT services firm Capgemini, two thirds of organizations are now prioritizing AI within their security operations.[1]
- The World Economic Forum's Global Cybersecurity Outlook 2025 reveals that 66% of organizations expect AI to significantly impact cybersecurity. Yet only 37% have processes to evaluate the security of AI systems before deployment.[2]
- A recent Bain & Company survey reports generative AI adoption is soaring, with 95% of US companies using it.[3]

This whitepaper explores the use of AI systems through three interconnected lenses: securing generative AI applications, using generative AI to strengthen overall security posture in the cloud, and protecting against generative AI-powered threats (Figure 1).

Figure 1 – Key considerations for generative AI security

Drawing from real-life insights and experience from building and operating cloud services, we'll share best practices, the power of automated reasoning to verify the security and correctness of complex systems, and key action items that can help you scale existing practices with generative AI to build a strong and sustainable security and compliance posture.

Understanding AI security and AI safety

The terms AI security and AI safety are often conflated. They are distinct but connected aspects of the development and deployment of AI systems.

- **AI security** primarily revolves around protecting AI systems from unauthorized access and tampering to maintain confidentiality, integrity, and availability. It acts as a shield against deliberate attempts to subvert, manipulate, or exfiltrate data.
- **AI safety** involves broader considerations related to developing and using AI in a way that maximizes its benefits to humanity and minimizes potential harm. It addresses unintended behaviors and system flaws, and provides fine-tuning and guardrails that increase the probability of ethical and reliable operation.

Together, they work to establish a solid foundation for responsible AI, which we will discuss later, and build societal trust.

The generative AI difference

While classical machine learning is typically used to make predictions or classifications based on existing data, generative AI can create new content and ideas. The technology offers the potential to help organizations innovate and enhance customer experiences, increase productivity, and optimize operations.

Although generative AI solutions create new pathways for data to flow through systems, security doesn't need to be reinvented to account for generative AI. Organizations that already have robust security and compliance infrastructure in place can accelerate generative AI adoption. A solid foundation in security fundamentals such as identity and access management, log monitoring, data protection and classification, security policy enforcement, incident response planning, and patching can help you adapt and expand your security program, and apply those capabilities to the nuances that make generative AI unique.

Maximizing the benefits of generative AI while mitigating its risks requires an approach that addresses the security of generative AI applications, the use of AI to enhance existing security practices, and protection against AI-powered threats.

Securing generative AI

Protecting generative AI systems from threats that could compromise their functionality or trustworthiness is critical. If the AI itself is vulnerable, this will undermine its effectiveness and value as a business tool. This is especially critical when AI is used to enhance security. Developing an understanding of how generative AI applications work, scoping your use cases and responsibilities, and following best practices can help you assess and implement security controls throughout the AI lifecycle.

Generative AI applications are composed of multiple interconnected components that work together to deliver intelligent capabilities (as shown in Figure 2). At the core of generative AI are machine learning models known as foundation models (FMs), which are trained using vast amounts of data. A large language model (LLM) is a type of FM that focuses specifically on generating outputs such as human language, source code, and configurations for computer programs. A large multimodal model (LMM) is another type of FM that processes and generates sound, images, and video streams. FMs provide the base intelligence for generative AI through model inference—the process the model uses to evaluate and analyze inputs (also called prompts) and generate appropriate outputs or responses.

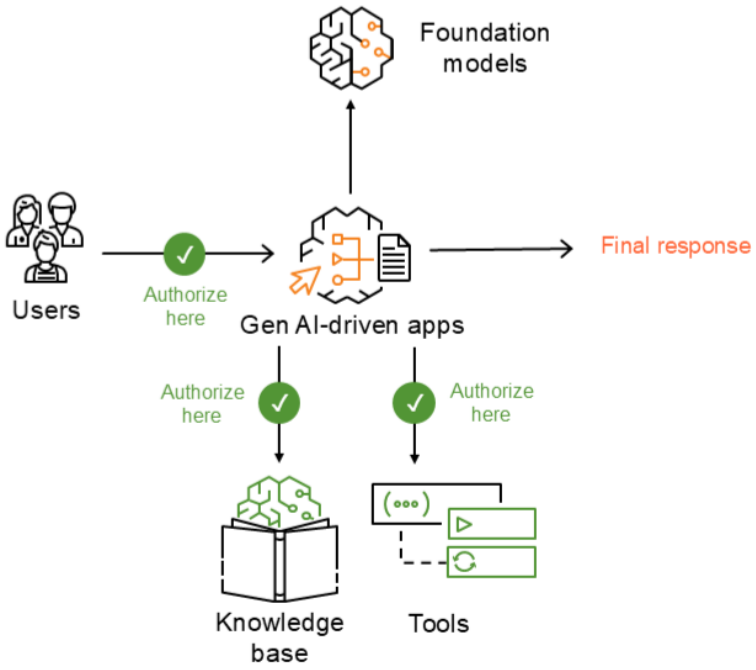

Figure 2 – Components of a generative AI application

Generative AI-powered agents go beyond basic query and response systems by using FMs to help orchestrate interactions between FMs, data sources, and external tools. Agentic AI systems offer a different approach compared to traditional software, particularly in their ability to handle complex, dynamic, and domain-specific challenges. While traditional systems rely on rule-based automation and structured data, agentic systems use FMs and agentic frameworks to operate in a more human-like, autonomous fashion. Agents learn from interactions with users, break down complex tasks, and coordinate multiple steps to achieve desired outcomes.

> Gartner® predicts that: "By 2028, 33% of enterprise software applications will include agentic AI, up from less than 1% in 2024," and "at least 15% of day-to-day work decisions will be made autonomously through agentic AI, up from zero percent in 2024."[4]

In the security domain, this shift from AI that *answers* to AI that *acts* (Figure 3) has the potential to help address alert fatigue and burnout by handling repetitive tasks at scale so that security teams can accomplish more, respond faster, and focus on strategic issues.

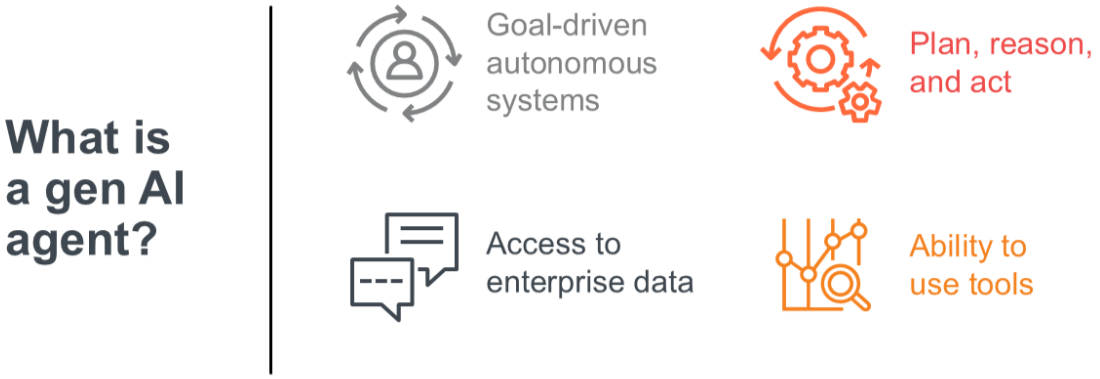

Figure 3 – Generative AI agents

However, as organizations start building agentic AI systems, addressing the risks that come with them is critical. An agent needs to be properly constrained to carry out its tasks based on the correct levels of authorization, and any additional permissions that are granted to it when acting on behalf of its caller (person or agent). Agents can invoke application programming interfaces (APIs) or tools, query knowledge bases, chain together multiple model interactions, and orchestrate interactions with other agents. This requires careful consideration around user and machine identity, tool access, action permissions, and maintaining appropriate security context throughout the workflow.

Scoping use cases and responsibilities

Establishing security requirements and the scope of your organization's responsibilities is a foundational step to generative AI adoption. Consider your use case: How will you interface with the application? What type of data are you using with the application? Is it accessible to your employees, or to customers? What questions do you not want the application to respond to? What identity and related permissions should it pass to tools and other agents?

The spectrum of implementations ranges from consuming existing AI services through public APIs, web user interfaces, and mobile applications, to building and training custom AI models from scratch (see Figure 4).

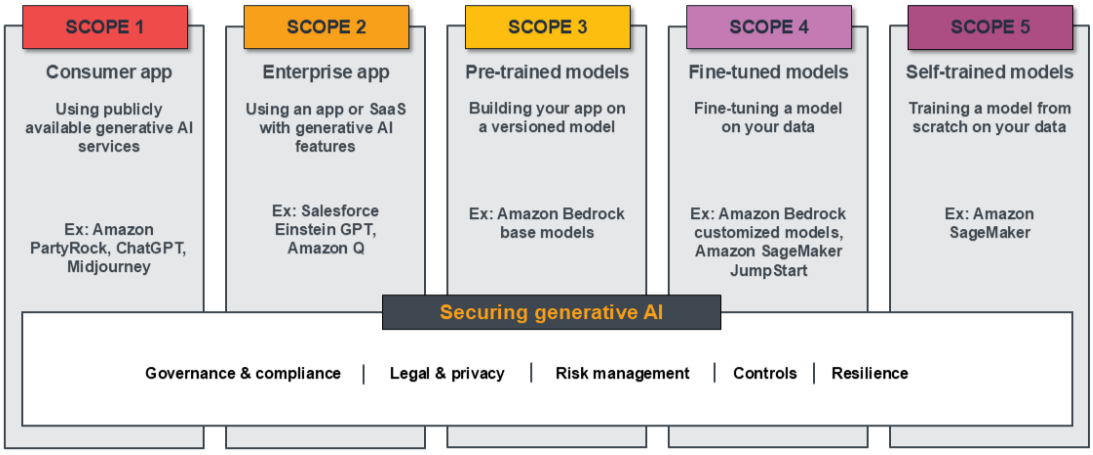

Figure 4 – Classifying use cases with a generative AI security scoping matrix

If you have a buyer's mindset (scopes one and two in Figure 4) and will be using an off-the-shelf application that's available to the public, your privacy, governance, and risk management concerns will be different than those associated with a builder's mindset (scopes three through five).[4] Identifying your scope will determine how you should implement security controls and drive your secure development process.

The shared responsibility model (SRM) delineates security roles across model providers, cloud providers, and organizations. It shifts based on implementation approach. Organizations consuming AI capabilities through public interfaces or applications typically have reduced responsibilities for the security of underlying AI infrastructure and models.[5] However, data handling, access controls, and compliance requirements must still be carefully considered when integrating these services.

Organizations building custom AI applications using foundation models, or training their own models, take on significantly more security responsibility across the entire stack. This includes securing training data, protecting model artifacts, implementing appropriate access controls, and maintaining the security of the infrastructure hosting these capabilities. Understanding these scope-based responsibility differences is critical to properly assessing risks and implementing appropriate controls.

Critical concepts

Three concepts can help you securely design generative AI solutions by breaking down complex ideas into understandable and usable chunks.

- **LLMs are functionally non-deterministic systems.** Identical inputs can produce different outputs because of sampling methods, *temperature* settings, and some inherent randomness in the generation process. This variability means security controls cannot rely on consistent model responses, and critical operations require external validation rather than depending on reproducible LLM behavior.
- **LLMs process inputs with equal privilege.** No security boundaries exist within the model. System prompts, retrieved documents, tool outputs, and user inputs become undifferentiated tokens processed through the same attention mechanisms (in current state-of-the-art LLMs), making it architecturally impossible to implement authorization or access controls within the model itself.
- **LLMs are statistical pattern completion engines.** Outputs are generated based on training patterns, rather than through logical deduction or verification of truth. This is why LLMs can produce convincing but factually incorrect outputs, and why they often cannot maintain logical consistency across complex multi-step operations without external verification.

Together, these concepts—which should be periodically revisited as technology changes—reveal why security architecture patterns such as external security boundaries instead of internal controls, formal verification instead of trusting model outputs, and defense-in-depth strategies that assume the model itself cannot be a security enforcement point, are necessary.

Securing generative AI best practices

Best practices can help you confidently develop and deploy generative AI systems while protecting against evolving threats and vulnerabilities. It is important to remain flexible and structure systems and organizational processes to be responsive to changes in the threat landscape as generative AI continues to evolve.

1. **Take a secure by design (SbD) approach:** Build security into the entire lifecycle of the generative AI system, from initial design to deployment and ongoing monitoring. Incorporate SbD tactics such as securing the infrastructure foundation of system components, maintaining up-to-date software through patching, and using memory safe languages.[6] Frameworks such as The National Institute of Standards and Technology (NIST) *Secure Software Development Framework* (SSDF), and the accompanying NIST *Secure Software Development Practices for Generative AI and Dual-Use Foundation Models: An SSDF Community Profile* can support your efforts with high-level secure software development practices and recommendations.
2. **Implement comprehensive access controls:** Apply the principle of least privilege across all AI system components including models, data stores, endpoints, and agent workflows. Implement multi-layered access controls for retrieval augmented generation (RAG) architectures and establish unique identities for AI agents with proper authorization levels.
3. **Secure data and communication flows:** Protect all system component interactions with private networks, and encrypt network data. Implement comprehensive security measures such as input sanitization, secure prompt catalogs, and data access governance. Establish private network communications with solutions such as AWS PrivateLink while maintaining data integrity throughout processing and transmission.
4. **Monitor and audit events:** Track both control and data planes for application performance, workload quality, and security across your generative AI workloads. Monitor authentication attempts, potential exploits, access patterns, unusual behaviors, rate limiting, quota usage, and possible data leakage.
5. **Control AI system behaviors:** Establish guardrails and boundaries that govern how AI systems interact with data and execute workflows. Implement response validation through techniques such as keyword identification, automated reasoning (which we will explore in the next section), and human review when necessary, while using RAG architectures to limit data access with traditional access controls and improve response accuracy.

6. **Secure system and user prompts:** Maintain a managed prompt catalog with least privilege access and implement robust sanitization of user inputs and model outputs to help prevent or mitigate prompt injection risks. Create clear context boundaries in prompt templates and validate prompts through multiple techniques including keyword searches and LLM-as-a-judge approaches as an input filter.
7. **Prevent excessive agency:** Set clear permission boundaries for LLM requests and agent workflows to help prevent unauthorized actions beyond defined scopes. Limit agent access to only necessary systems and data sources, especially in automated processes. Create throttles and alarms on automated processes to mitigate the risk of automation loops or run-away tasks.
8. **Detect and remediate data poisoning risks:** Protect models you are training by isolating training environments and implementing thorough data cleaning processes before training. Use toxicity detection and evaluation techniques to help identify and limit potentially harmful data from entering the training process. If you are fine-tuning an existing LLM, maintain the security of your fine-tuning data.
9. **Conduct penetration testing:** Perform regular red team exercises to systematically probe both AI models and supporting systems for vulnerabilities and weaknesses. Automate penetration testing for continuous assessments. This practice helps develop AI systems that are functional and secure, while promoting trust in your organization's AI solutions.

The Open Worldwide Application Security Project (OWASP) Gen AI Security Project, a community-driven initiative focused on identifying, mitigating, and documenting security risks associated with generative AI technologies, offers resources that can facilitate your efforts. These include the *2025 OWASP Top 10 List for LLM Applications*, which facilitates the assessment and prioritization of critical security risks related to LLM applications, the *OWASP Agentic AI Threats and Mitigations* guide that provides a threat-model-based reference of emerging agentic threats, and the *OWASP Securing Agentic Applications Guide* that focuses on concrete technical recommendations that builders and defenders can apply directly.[7]

The role of automated reasoning

LLMs can generate human-like text by encoding statistical patterns and relationships from vast amount of textual data during the training phase. This use of patterns and relationships can lead to outputs that sound plausible but are factually incorrect or misleading, a phenomenon known as hallucination. While hallucinations aren't inherently *bad* in all scenarios—and they reflect the

creative potential of LLMs—they can produce incorrect output when correctness is expected or required with certain use cases. Supplementing generative AI with techniques based on formal logic is necessary to verify correctness.

This is where automated reasoning comes in. Automated reasoning, also known as formal verification, reasons through symbol manipulation using sound logical rules to construct proofs. It's the same approach to building proofs that you might have learned in high school geometry, scaled up with algorithms and tooling to enable reasoning about complex digital systems such as communication protocols, user authorization, memory safety, and functional correctness of code. In the context of generative AI, automated reasoning can act as a guardrail that helps prevent correctness errors and security vulnerabilities using logically accurate and verifiable reasoning that filters AI responses or generated code to help ensure they are either correct or rejected because they do not pass the filter provided by the proof.[8]

Proof automation

Well-known mathematical concepts, such as the Pythagorean theorem that describes the relationship between the three sides of a right triangle, demonstrate the importance of proofs. The theorem states that the square of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. This can be written as $a^2 + b^2 = c^2$ where *c* is the length of the hypotenuse while *a* and *b* are lengths of other two sides. After this truth was proven by Euclid in 300 BC, it was used as a mathematical building block for precise measurements, facilitating advancements in architecture, construction, and navigation. It also set the foundation for more complex mathematical proofs and has applications in fields ranging from physics to game development.

Consider how we established the truth of the Pythagorean theorem. By looking at a large (but finite) number of right triangles, we can see a relationship between the length of their sides a, b and c, as mentioned previously. The problem is, there are a potentially infinite number of right triangles; enumerating and checking that the theorem holds for each of them would also take an infinite amount of time. Instead, mathematicians used logical rules and a few proof steps to reach the sound conclusion that all right triangles satisfy this property. Automated reasoning takes this approach and scales it by using algorithms to assist the proof engineer in building and maintaining proofs of complex digital systems as they evolve.

Key differentiators

Automated reasoning takes a different approach than machine learning and traditional software testing.

- **Machine learning:** Machine learning takes large amounts of data—such as many right triangles with sides that have respective values of a, b and c—and learns patterns during training. The training process could result in a model that distinguishes right triangles from other triangles based on the length of their sides, or predicts one side given the two other sides, honoring the Pythagorean theorem ($a^2 + b^2 = c^2$). While the machine learning model in this example can distinguish right triangles from other triangles with a low rate of error, it might never be perfect. It will sometimes misclassify a triangle or wrongly predict the length of the hypothenuse. In short, machine learning makes predictions and inferences that depend on the quality and quantity of the data it's trained on. Automated reasoning techniques, on the other hand, provide proof. Instead of training a model with data, they deduce an outcome based on mathematical logic.
- **Traditional software testing:** Testing methods such as design reviews, code audits, stress testing, fuzz testing, and fault injection focus on validating system behavior under specific scenarios, whereas automated reasoning aims to use logic to verify system behavior under any possible scenario (see Figure 5).

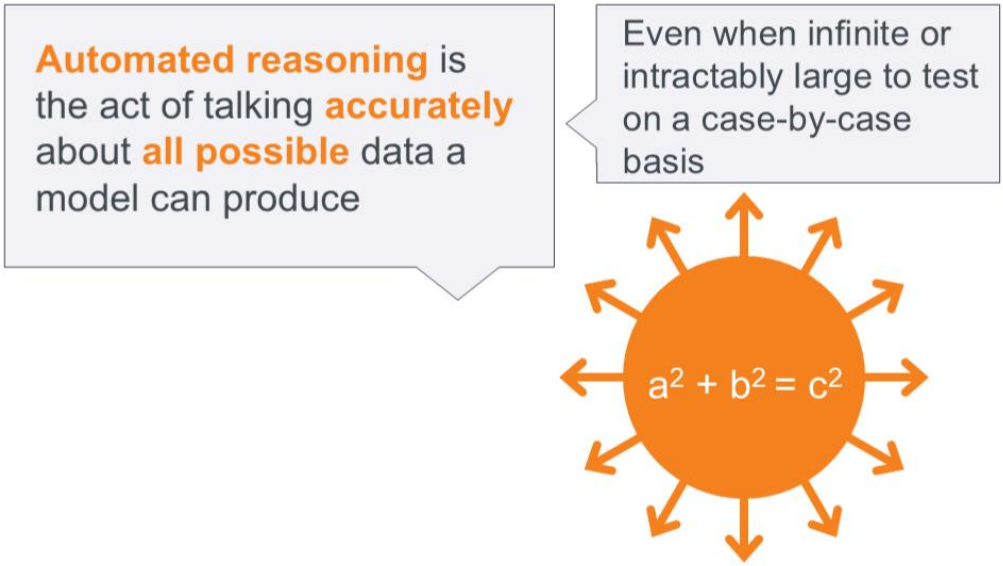

Figure 5 – Automated reasoning

How it works

Automated reasoning is particularly helpful for use cases where sound reasoning and explainability are important. When you use automated reasoning, you first build a formal model of your domain, in mathematical logic, from the specification of the system. Then you document assumptions about your environment in mathematical logic. All models use assumptions about the environment of the system being modeled, but that does not limit the models' use. Next, you present the automated reasoning subsystem (implemented as a software service or tool) with a problem statement to determine/prove if an assertion about the system is always true (valid), always false (invalid), or sometimes true/sometimes false (conditionally valid) according to your model.

Consider the example of a generative AI-based chatbot that answers questions for airline customers. The following policy forms the basis for the query:

> If any segment of a ticket is canceled by the airline for operational reasons (and not due to weather) the customer is entitled to a full refund.

Using this human-readable policy, a mathematical model is created that captures the condition for refund.

Now assume a customer (represented by C_1) has a ticket (represented by T_1) that includes a flight segment (represented by S_1), and the airline canceled that segment for non-weather-related reasons.

Customer C_1 asks the chatbot, "Am I entitled to full refund?" and the chatbot, which relies solely on output from its underlying AI model, answers *yes*. Given that the AI system is based on a probabilistic model with no deductive logic abilities, how can the airline organization verify that this answer is correct? Automated reasoning can be used behind the scenes as a guardrail (built into the chatbot application) to evaluate the AI response, using the above-mentioned symbolic model. Specifically, it uses logical deduction as a proof technique. In this case:

- Customer C_1's ticket T_1 includes segment S_1 = TRUE AND
- Airline cancelled segment S_1 = TRUE AND
- Cancellation of S_1 is not due to weather = TRUE.

Because *all* conditions are true, automated reasoning would verify that the AI model is correct in answering *yes* to the customer's question using the chatbot application.

Now consider a situation in which the customer (represented by C_2) has a ticket (represented by T_2) that includes a flight segment (represented by S_2) that was canceled due to weather.

The customer asks the chatbot if they're entitled to a full refund, and the AI model responds with *yes*. Automated reasoning system can once again evaluate if the answer is valid:

- Customer C_2's ticket T2 includes segment S_2 = TRUE AND
- Airline cancelled segment S_2 = TRUE AND
- Cancellation of S_2 is not due to weather = FALSE

Because some conditions aren't met, automated reasoning would catch the AI model's error and return an *invalid* status to the calling application. The chatbot application will not use AI's output in this case to respond to the customer.

Business policies can have millions of such conditions with interdependencies. By modeling a system's properties and requirements in mathematical logic and reasoning soundly about them, automated reasoning helps efficiently and authoritatively answer critical questions about the system's future behavior. Used alongside other techniques such as prompt engineering, RAG,

and contextual grounding checks, automated reasoning-based guardrails add a rigorous approach to verifying that LLM-generated output is logically consistent with the domain model.

Beyond binary verification with yes/no answers, automated reasoning techniques can also help with more complex use cases such as solving constraint satisfaction problems and offloading logical deductions from LLMs.

Security use cases

The combination of automated reasoning and generative AI can help you accelerate security efforts in security-critical use cases:

- **Code generation:** AI-generated code can be automatically verified for security properties using automation reasoning-based program analysis tools. Memory safety, absence of buffer overflows, proper input validation, and other properties can sometimes be mathematically proven. Examples of languages that have a growing framework of formal verification tools include Rust, with code-verifying tools such as Kani, AutoCorrode, and Verus, and the verification-aware Dafny language.
- **Infrastructure configurations:** AI-generated cloud configurations, network topologies, and deployment scripts can be verified against security policies using specialized automated reasoning tools that understand infrastructure semantics. Examples of reasoning tools for virtual private cloud (VPC) configurations include the AWS Reachability Analyzer, powered by underlying reasoning engines, which can check the reachability properties of a VPC configuration and certify that no private database in your VPC can be accessed from the public internet.

- **Authentication and authorization policies:** AI systems that generate identity and access management policies, role-based access control configurations, or authentication flows can have outputs verified for consistency, completeness, and compliance with security principles using automated reasoning. For example, Cedar, an open source authorization policy language used by Amazon Verified Permissions, can be used in conjunction with AI to assure developers that authorization decisions will be correct. Cedar's trustworthiness is based on a process called verification guided development that uses automated reasoning.[9] Using Cedar, developers define who (the *principal*) can do what (the *action*) on what target (the *resource*) under which conditions (*when*). AI software development assistants such as Amazon Q Developer can generate Cedar policies based on a developer's natural language input such as "modify the policy to ensure principal A can access resource R," simplifying authorization without compromising correctness.
- **Compliance:** Security assurance and compliance teams are routinely asked to answer due diligence questions. LLM-based RAG architecture can be used to retrieve the most relevant sections or paragraphs from a vetted knowledge base, and produce a succinct answer based on the selected content. These LLM-generated answers can be more useful than raw sections that tend to be long walls of text; however, they need to be reviewed for inaccuracies or hallucinations. Automated reasoning can help you build a semantic model of the knowledge base first and validate AI-generated answers against the model.

Considerations

Just like machine learning, which relies heavily on the quantity and quality of training data and lacks human-like reasoning abilities, the use of automated reasoning requires some effort and caution.

- **Assumptions:** While automated reasoning promotes a higher level of confidence in correctness than is possible by using traditional software development and testing methods, it depends on making assumptions about the environment of the system being modeled or reasoned about. For example, the model of a system might incorrectly assume that underlying components such as compilers and processors don't have any bugs (although it is possible to formally verify those components as well). End-to-end testing remains crucial to validating these assumptions, so that proofs can be updated as necessary.

- **Ambiguity:** Automated reasoning provides definitive answers based on sound reasoning rules. It cannot handle ambiguous specifications that can't be expressed in mathematical logic. However, the efforts made to create a model of a system will often reveal those ambiguities and allow them to be corrected, which is a win-win for the system and its provability.
- **Expertise:** Applying automated reasoning to complex systems can require significant mathematical logic expertise.

The convergence of AI and automated reasoning represents a fundamental shift toward building systems that are both powerful and provably correct. Generative AI provides flexibility and natural-language-based interaction, while automated reasoning facilitates mathematical certainty and security.. Additionally, while automated reasoning helps improve the trustworthiness of generative AI, generative AI provides benefits to automated reasoning by making it more accessible, helping non-expert users elicit, understand, and formalize their requirements as a first draft model that can be quickly reviewed, refined and processed by automated reasoning tools. As we move toward autonomous agentic AI systems, the combination of generative AI and automated reasoning will become increasingly important to building secure and reliable systems at scale.

Using AI for security, and protecting against AI-powered threats

Current security processes require significant human involvement to maintain the security and safety of software products and services. While automation is being used to scan environments and repositories—and advancements have been made in fields such as automatic patching and automatic remediation of code and configuration—these tools require human oversight. Core tasks such as application security reviews, penetration testing, and incident response remain inherently human-driven, and resource needs often increase alongside business growth, creating scaling issues. The emergence of AI-powered threats presents additional challenges, lowering the barrier of entry for malicious actors. Adversaries can now launch threats faster, with a higher degree of sophistication, at a reduced cost. Investing in generative AI and machine learning for security, particularly with an eye towards enhancing conventional automation systems, can help you address threats and bolster the efficiency of security teams in several areas.

Application security

Traditional application security processes, practices, and tools are evolving to include the new capabilities that generative AI provides. AI-powered security agents can help you automate routine tasks and significantly reduce development time and costs by providing real-time security feedback to developers, facilitating automated penetration testing, and generating code for security fixes. They can also interact with traditional tools such as static application security analysis (SAST) and software composition analysis (SCA) using Model Context Protocol (MCP)—an open source standard for connecting AI applications to external systems. Additionally, generative AI-powered software development assistants such as Amazon Q Developer and agentic integrated development environments (IDEs) such as Kiro can accelerate the development of custom security tooling and automated security tests by turning natural language prompts into clear requirements, system design, and discrete tasks.

Threat detection and analysis

As systems and architecture evolve, so do adversarial tactics, techniques, and procedures (TTPs). Continuously researching exploit scenarios, modeling them in infrastructure, and working backwards to create detection rules is essential. Generative AI can be used to partially automate research, the creation of detection rules, and validation steps so that you can detect and respond to threats faster and use fewer resources in the process.

While individual AI tools excel at specific tasks—such as network traffic analysis, domain reputation scoring, or behavioral analytics—combining multiple specialized AI systems creates a more comprehensive security posture. A multi-layered approach allows organizations to correlate insights from disparate tools. For example, bringing AI-powered network analysis together with domain reputation scoring and user behavior analytics can ease the identification of sophisticated threats that might not be apparent when looking at a single data source. The Open Cybersecurity Schema Framework (OCSF), a collaborative, open source effort led by AWS and partners across the cybersecurity industry, can help your security team work with a common language for threat detection and investigation.

The OCSF addresses the absence of inconsistent formats and data models for logs and alerts with a vendor-agnostic schema that simplifies data ingestion, normalization, and analysis across different security tools for security log producers and consumers.[10] This standardization creates a unified foundation for advanced analytics and AI-powered tools. With OCSF-formatted data,

you can use generative AI to enhance your security operations in multiple ways. For example, generative AI can analyze OCSF-formatted security events to automatically map activities to MITRE ATT&CK® TTPs, enhancing investigation capabilities with contextualized insights.[11]

Security operations

Effective security operations are critical for detecting, preventing, and responding to threats. The use of generative AI can enhance security operations by creating comprehensive narratives through finding group summaries, offering context-rich overviews of security incidents across multiple sources. This approach helps identify threats that might be missed when analyzing isolated insights, improving investigation efficiency and response times. You can use generative AI to augment your team's efforts in several ways:

- Automate alert triage in security information and event management (SIEM) and security orchestration, automation, and response (SOAR) systems
- Provide real-time guidance during security operations center (SOC) investigations
- Support analyst training through integration with existing tools
- Promote secure practices while reducing security team workload

This combination of human insight and AI tools helps optimize threat identification and response and automate routine tasks while maintaining appropriate operational boundaries and controls.

Vulnerability management

Security teams often spend significant time on common vulnerabilities and exposures (CVE) analysis, enriching information for internal teams and customers with additional metadata using the Common Vulnerability Scoring System (CVSS), and the Common Weakness Enumerations (CWE) taxonomy. Proper CVSS scoring considers factors such as the complexity of the vulnerability, its impact on confidentiality, integrity, and availability, and the potential for exploitability. In addition to the CVSS score, the CVSS vector string that represents a vulnerability's characteristics encodes information about the attack vector, threat complexity, privileges required, user interaction, and scope. Analysis of this detail is a complex and time-consuming process that requires weeks of training and practice. A trained security engineer typically spends over an hour analyzing a single CVE. A generative AI-based CVE analyzer

application that is contextually grounded using a knowledge base that contains prior analysis data can reduce analysis time from hours to seconds, while maintaining output consistency.

The importance of defense in depth

Taking a defense-in-depth approach can support your efforts to secure generative AI applications, use generative AI to strengthen your security posture, and protect data and systems against AI-powered threats. Defense-in-depth uses multiple layers of defense (shown in Figure 6) that are built upon policies, procedures, and awareness that establish a solid security baseline and security culture. This layered approach—which includes threat detection and incident response, data protection, application protection, network and edge protection, infrastructure protection, and identity and access management defenses—is crucial for limiting the extent of damage or compromise if a single security control is breached.

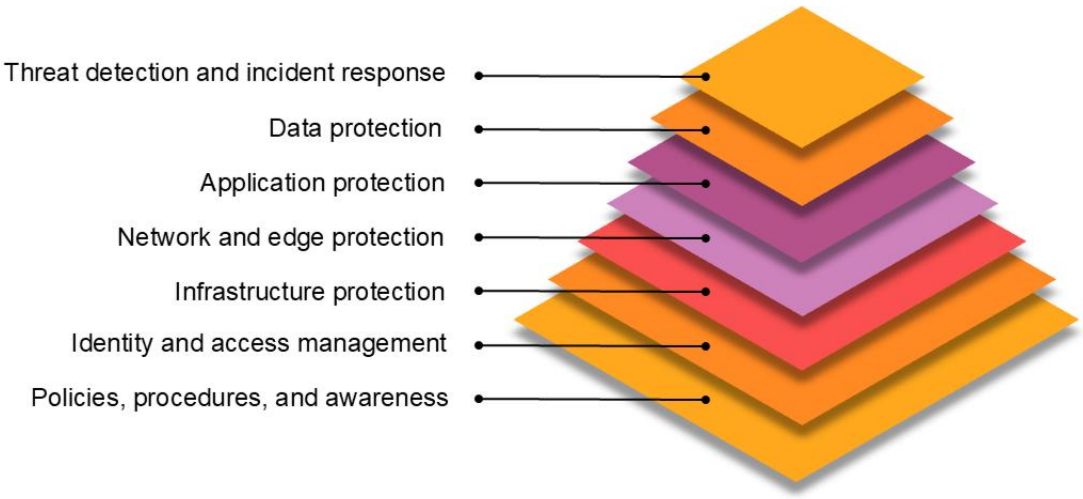

Figure 6 – Defense in depth security

The effectiveness of this strategy depends on how seamlessly the components work together. For example, when implementing agentic AI systems, identity controls must integrate with infrastructure protection to maintain security context throughout automated workflows. It's important to view the defensive layers not as independent barriers, but as an interconnected security framework in which each layer reinforces the others. If one security control fails, others can continue to protect the system, making it more difficult for threat actors to compromise applications while empowering you to innovate with a secure foundation.

Zero trust complements defense in depth with a security model and associated set of mechanisms that focus on providing security controls around digital assets that don't solely or fundamentally depend on traditional network controls or network perimeters. Zero trust encourages you to incorporate a wide range of context about any particular access request, including identity, device, data, behavior, and more, so your systems can make increasingly granular, continuous, and adaptive policy-based access control decisions.[12]

AI upskilling

Effectively using generative AI for security and protecting against generative AI-powered threats requires both advanced defenses, and skilled professionals. Upskilling is critical—without continuous training, security teams risk falling behind.

Create a learning culture that allows for hands-on experience and experimentation with AI tools in daily work. Emphasize the importance of communication and transparency. Sharing success and failures within and across teams can facilitate your efforts to boost morale, and to use wins and losses to strengthen your organization's security posture.

The following focus areas can help you build an effective upskilling strategy:

- Collecting feedback from employees about the AI skills they want and need
- Aligning learning efforts with security and business objectives
- Setting realistic and measurable goals
- Tailoring learning with approaches that range from on-the-job training and mentorship programs to self-paced learning solutions and instructor-led workshops
- Conducting regular skills audits and gap analyses
- Measuring employee engagement and satisfaction

Specialized AI training and resources from AWS and providers such as SANS Institute can support your upskilling efforts.[13, 14]

Governance and compliance considerations

The global AI regulatory landscape is evolving at an unprecedented pace. From the comprehensive risk-based framework provided by the European Union Artificial Intelligence Act (EU AI Act) to state-level laws in the US, organizations are navigating increasingly complex

compliance challenges. As the patchwork quilt of rules and regulations continues to grow, it's important to integrate both security and compliance considerations into your organization's AI development and deployment plans. This will help you adapt to emerging requirements while maintaining consistent security and governance controls.

> AI governance refers to the frameworks, policies, and organizational processes that focus on aligning AI development and deployment with business objectives, regulatory requirements, and ethical standards.

Rather than chasing compliance with specific regulations, focus on the principles of responsible AI. At AWS, for example, AI innovation is guided by eight dimensions shown in Figure 7 and described in the following list.[15]

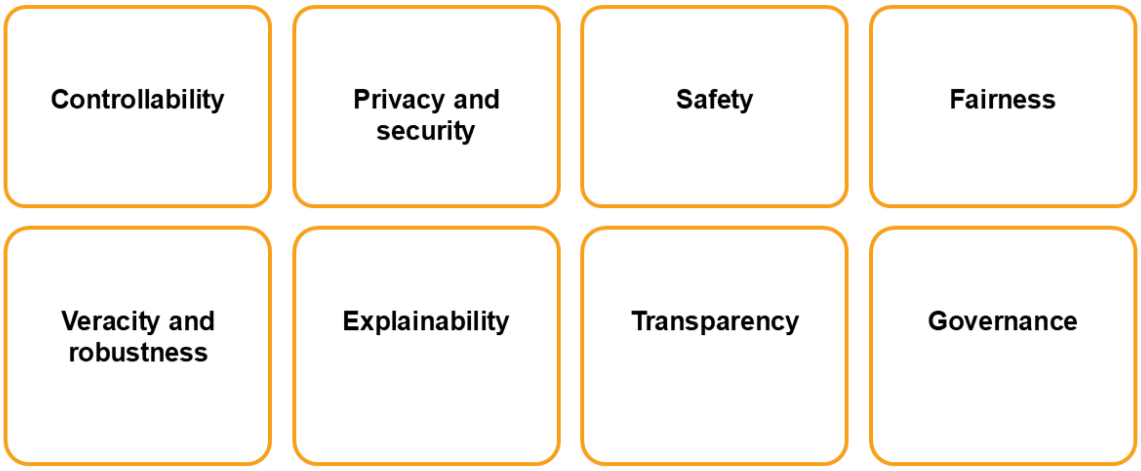

Figure 7 – AWS core dimensions of responsible AI

- **Controllability:** Having mechanisms to monitor and steer AI system behavior.
- **Privacy and security:** Appropriately obtaining, using and protecting data and models.
- **Safety:** Preventing harmful system output and misuse.
- **Fairness:** Considering impacts on different groups of stakeholders.
- **Veracity and robustness:** Achieving correct system outputs, even with unexpected or adversarial inputs.

- **Explainability:** Understanding and evaluating system outputs.
- **Transparency:** Enabling stakeholders to make informed choices about their engagement with an AI system.
- **Governance:** Incorporating best practices into the AI supply chain, including providers and deployers.

Implementing responsible AI practices throughout the AI lifecycle and adopting voluntary standards such as the *NIST AI Risk Management Framework (AI RMF)* and *ISO/IEC 42001:2023* can help you stay ahead of regulatory actions as you innovate and map existing controls to new requirements rather than building compliance programs from scratch.[16]

Several best practices can facilitate your AI governance and compliance efforts:

- **AI discovery and inventory:** Build and maintain a comprehensive AI inventory that goes beyond obvious applications such as chatbots or recommendation engines to identify AI components embedded within your traditional applications, third-party AI services, and AI systems within your supply chain. Many organizations are surprised to discover the extent of AI usage within their commercial software packages and data processing pipelines. Build out sufficient observability to address shadow generative AI. AI monitoring dashboards can help you track violations of corporate AI policies. Endpoint monitoring solutions should be deployed to detect unauthorized use.[18]
- **Risk assessments:** Conduct AI risk assessments—traditional risk assessments that focus primarily on cybersecurity, data protection, and system availability are important, but AI systems introduce new categories of risk that must be evaluated. These include societal impact and potential for harm, bias, and fairness implications, and the challenges of model drift and accuracy degradation. The potential for excessive autonomous behavior and the need for transparency and explainability add additional layers of complexity. These considerations must be evaluated alongside traditional risk factors to form a comprehensive foundation for compliance decisions.
- **Policies and procedures:** Establish an organization-wide policy to provide requirements for building and using generative AI services in accordance with data classification, security, and legal policies. Provide clear communication to stakeholders at all levels to support and build consistent control implementation.

- **Documentation:** Thoroughly document decisions, processes, and controls. It is important to maintain clear records of AI systems, including risk assessments, development procedures, and ongoing monitoring protocols. This can help your organization demonstrate compliance, provide operational consistency, and strengthen incident response. Continuous security monitoring controls such as model behavior tracking, input validation checks, and output verification should also be documented alongside traditional compliance measures and integrated into existing security frameworks.
- **Roles and responsibilities:** Define roles and responsibilities for AI oversight, from executive sponsorship to technical leadership and operational management. Understanding shared responsibility in AI compliance is also imperative. Establish a clear understanding of your obligations in addition to those of your service providers, model developers, and other partners. Security responsibilities should be clearly delineated between parties, with specific requirements for incident response, vulnerability management, and security testing documented in service level agreements. This understanding should inform contractual agreements and internal policies, making certain that all parties know their role in maintaining compliance.
- **Continuous monitoring:** Stay informed about regulatory changes and regularly assess your AI systems for compliance with requirements and responsible AI principles.

The implementation of an AI governance and compliance program should follow a measured, phased approach. Start by establishing system inventory and risk assessment processes. From there, you can develop core policies based on responsible AI principles, implement governance structures, and deploy monitoring systems to help detect unauthorized use of generative AI. Create feedback loops that enable continuous improvement and adaptation to emerging requirements rather than relying on point-in-time monitoring and assessments. The goal is to remain adaptable while adhering to core principles of responsible AI development and deployment, promoting compliance as an enabler of innovation rather than a barrier to progress.

Balancing AI automation with human oversight

Striking the right balance between AI automation and human oversight is central in the effort to keep fundamental security principles at the core of your security practices, while using automation to verify and enforce these principles at scale.

Clear boundaries between what AI can handle autonomously and where human judgment is essential need to be established. For routine, well-defined tasks with limited potential for harm, you might be comfortable with higher degrees of AI autonomy. For instance, AI can independently analyze network traffic patterns, classify potential threats based on known signatures, and even automatically block certain types of malicious activities that follow established patterns. However, when it comes to decisions with significant potential consequences, human judgment remains essential. At Amazon, for example, we don't allow AI to autonomously shut down critical services, make major configuration changes to production systems, or take actions that could substantially impact customer experience without human verification.

A helpful principle to follow is that the higher the potential impact of a decision, the more human oversight should be required. When AI systems detect what appears to be a sophisticated threat pattern at Amazon, we assemble the evidence and present it to security engineers with a recommendation; the decision to act rests with human experts. This approach recognizes that while AI excels at pattern recognition and processing vast amounts of data, humans provide unique capabilities—including contextual understanding, creative problem-solving, and judgment in ambiguous situations. The most effective security comes from combining these strengths rather than trying to replace one with the other.

Three key action items

Three key action items can help set your organization on the path to successfully securing generative AI applications, using generative AI to enhance security, and protecting against generative AI-powered threats.

- **Securing generative AI:** Determine the scope of your generative AI implementation and the responsibilities shared between your organization and service providers. Will you be consuming existing AI services through public APIs and mobile applications, or are you building and training custom AI models from scratch? Your use case will determine how you should implement security controls and drive your secure development process.
- **Using generative AI for security:** Identify areas where AI can effectively augment your existing security processes, without compromising reliability. Focus on automating routine tasks such as ticket management while maintaining human judgment for critical security decisions. Facilitate proper integration with existing security tools and workflows through standardized formats such as OCSF.

- **Protecting against generative AI-powered threats:** Monitor emerging AI-generated attack vectors and TTPs and update your threat models and security controls accordingly. Implement AI-based detection and response capabilities that can match adversarial sophistication and scale, while facilitating appropriate human oversight for critical security decisions.

Conclusion

As AI continues to transform how we innovate, balancing its opportunities and challenges is imperative. Effectively securing generative AI applications, using generative AI to accelerate security practices, and defending against generative AI-powered threats is an iterative process. Success requires a flexible approach that can help your teams maneuver through security and compliance changes as technologies evolve. Secure adoption might seem daunting, but scoping your use cases and responsibilities, following security best practices, and establishing strong governance frameworks can help you implement appropriate controls and advance your organization's security posture with AI while managing risks. Forging a dynamic human-AI partnership is key. By combining the speed and scalability of AI with human judgment and oversight, you can create more robust and adaptable security programs.

Contributors

This whitepaper was written by the following authors:

- Paul Vixie, VP, Distinguished Engineer, and Deputy CISO, AWS
- Debashis Das, Principal, Office of the CISO, AWS
- Riggs Goodman, Principal Partner Solution Architect, AWS
- Brandon Evans, Senior Instructor, SANS Institute

The authors would like to thank the following people for their insight and contributions.

- Albin Vattakattu, Security Engineer, AWS Security
- Anne Grahn, Senior Worldwide GTM Specialist, AWS Wickr
- Benjamin Dynkin, Principal, AI Assurance, AWS Security
- Byron Cook, VP and Distinguished Scientist, Agentic AI Leadership, AWS
- Clarke Rodgers, Senior Principal, AWS OCISO

- Danielle Ruderman, Worldwide Specialist Leader, AWS WWSO Security Services
- Hart Rossman, VP, Security & Infrastructure, AWS Global Services Security
- Jason Garman, Principal Security SA, AWS
- Mark Ryland, Senior Principal, AWS Security
- Matt Saner, Senior Manager, Security SA, AWS
- Nico Rosner, Senior Applied Scientist, AWS Provable Security
- Remi Delmas, Principal Applied Scientist, AWS Agentic Automated Reasoning
- Segolene Dessertine-Panhard, Senior Research Science Manager, AWS Generative AI Innovation Center
- Serdar Tasiran, Principal Applied Scientist, AWS Agentic Automated Reasoning
- Stanislas Prehu, Senior Regulatory Specialist, AWS Security Assurance
- Tancrede Lepoint, Principal Applied Scientist, AWS Provable Security

Document revisions

Date	Description
November 2025	First publication

Notes

[1] "New defenses, new threats: What AI and Gen AI bring to cybersecurity," https://www.capgemini.com/insights/research-library/generative-ai-in-cybersecurity?utm_source=pr&utm_medium=referral&utm_content=cybersecurity_none_none_pressrelease_none&utm_campaign=cybersecurity_ai/gen_ai_in_cybersecurity

[2] "Global Cybersecurity Outlook 2025," https://www.weforum.org/publications/global-cybersecurity-outlook-2025/digest/

[3] "Bain Generative AI Survey," https://www.bain.com/insights/survey-generative-ai-uptake-is-unprecedented-despite-roadblocks/

[4] Gartner Research, Top Strategic Technology Trends for 2025, By Gene Alvarez, Tom Coshow etc., Oct 2024. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.

[4] "Generative AI Security Scoping Matrix," https://aws.amazon.com/ai/generative-ai/security/scoping-matrix/

[5] "A secure approach to generative AI with AWS," https://aws.amazon.com/blogs/machine-learning/a-secure-approach-to-generative-ai-with-aws/

[6] "Building Security from the Ground Up with Secure by Design," https://d1.awsstatic.com/partner-network/AWS-SANS-Secure-by-Design-Whitepaper-2024.pdf

[7] "OWASP GenAI Security Project," https://genai.owasp.org/

[8] "Minimize AI hallucinations and deliver up to 99% verification accuracy with Automated Reasoning checks: Now available," https://aws.amazon.com/blogs/aws/minimize-ai-hallucinations-and-deliver-up-to-99-verification-accuracy-with-automated-reasoning-checks-now-available/

[9] "How we built Cedar with automated reasoning and differential testing," https://www.amazon.science/blog/how-we-built-cedar-with-automated-reasoning-and-differential-testing

[10] "Open Cybersecurity Schema Framework," https://github.com/ocsf

[11] "Powering AI-Driven Security with the Open Cybersecurity Schema Framework," https://aws.amazon.com/blogs/opensource/powering-ai-driven-security-with-the-open-cybersecurity-schema-framework/

[12] "Zero Trust: Charting a Path to Stronger Security," https://aws.amazon.com/executive-insights/content/zero-trust-charting-a-path-to-stronger-security/

[13] "New AWS Skill Builder course available: Securing Generative AI on AWS," https://aws.amazon.com/blogs/security/new-aws-skill-builder-course-available-securing-generative-ai-on-aws/

[14] "AI Security Starts Here," https://www.sans.org/mlp/artificial-intelligence

[15] "Transform responsible AI from theory into practice," https://aws.amazon.com/ai/responsible-ai/

[17] "AI lifecycle risk management: ISO/IEC 42001:2023 for AI governance," https://aws.amazon.com/blogs/security/ai-lifecycle-risk-management-iso-iec-420012023-for-ai-governance/

[18] "Shadow Generative AI," https://docs.aws.amazon.com/whitepapers/latest/navigating-security-landscape-genai/shadow-generative-ai.html