Salesforce Security

Mitigating LLM Risks Across Salesforce's Gen AI Frontiers

Contents

Introduction

At Salesforce, our top priority is trust, which drives our commitment to providing the most secure, reliable, and available cloud computing services. Our data-centric approach supports our dedication to customer success and ensures the highest levels of availability, performance, and security.

The Einstein 1 Platform, is the next generation of Einstein that currently delivers more than 200 billion AI-powered predictions daily across the **Customer 360** suite of products. By combining Einstein models with leading large language models (LLM), customers can use natural-language prompts on Salesforce application data to trigger powerful, time-saving automation and generate personalized content.

The Einstein 1 Platform combines Salesforce proprietary AI models with generative AI technology from an ecosystem of partners and real-time data from Salesforce products (Sales Cloud, Service Cloud, Marketing Cloud, Commerce Cloud, Tableau) along with **Salesforce Data Cloud**, which ingests, harmonizes, and unifies all of a company's customer data. Customers can then connect that data to OpenAI's advanced AI models out-of-the-box, or choose their own external models.

200B The Einstein 1 Platform delivers more than 200 billion AI-powered predictions daily

> "With the rapid increase in Gen AI solutions, there's growing demand to secure these solutions.

Then, they can use natural-language prompts directly in their Salesforce applications to generate content that continuously adapts to changing customer information and needs in real time.

The core player of a Gen AI solution is the foundational LLM, and recognizing the importance of protecting this paradigm, the Open Web Application Security Project (OWASP) recently revealed the top 10 LLM risks. This document sheds light on how the Einstein 1 Platform addresses these risks, the risks and controls in place, and provides guidelines for security teams to use to evaluate Gen AI security.

Salesforce continues to be actively represented in the OWASP Top 10 for LLM **Core Team**, **NIST AI RMF Team**, and working group. This white paper aims to highlight how Salesforce's AI platform strategically addresses these risks with details on identified risks and control strategies. This is a valuable resource for understanding and mitigating potential threats. The objectives of the insights covered in this document, are to help security teams develop guidelines and empower them to assess & grade the security of Gen AI applications. As technology continues to evolve, this whitepaper aims to be a guiding light in the midst of the growing complexity of gen AI security.

LLM01 Prompt Injections

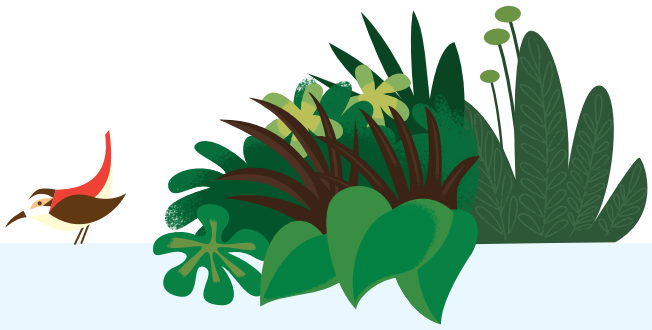
Risk Overview

Prompt injection vulnerability occurs when an attacker manipulates the operation of a trusted large language model (LLM) through crafted inputs. This results in the LLM acting as a "confused deputy" for the attacker. Given the high degree of trust associated with an LLM's response, the manipulated responses may go unnoticed and even be trusted by the user, which allows the attacker's intentions to take effect.

Prompt injection vulnerabilities arise from the inherent characteristics of LLMs, which lack a clear demarcation between instructions and external data. This stems from the fact that both instructions and external data are processed using natural language, blurring the distinction between user-provided input types. Consequently, in the LLM's internal workings, there's no certain method to prevent such attacks.

How Does Einstein Defend Against Prompt Injection?

The Einstein Trust Layer is where we strengthen our defenses against potential prompt injection attacks that target our LLMs and our Platform that consumes these AI-generated responses.



Our approach to safeguarding against these threats involves a two-pronged strategy: a mix of intelligent and deterministic defense mechanisms.

Deterministic Defenses

In the Einstein Trust Layer, Salesforce leverages several heuristic strategies recommended by the research community to safeguard against potential threats to prompts, including deny list-based filtering, instruction defense, random sequence enclosure, and various others.

Intelligent Defenses

Salesforce actively researches advanced machine learning (ML) driven defense strategies to intelligently detect and prevent various forms of malicious insertions within prompts. These cutting-edge models are integrated into our Trust Layer Gateway, enhancing the security of the Einstein 1 Platform.

For instance, Salesforce uses its proprietary ML models in the Trust Layer to prevent it from generating toxic content (hate, identity, violence, physical, sexual, profanity language). This ensures the Einstein 1 Platform aligns with the **ethical guidelines** and security requirements Salesforce expects.

Here are specific examples of how our Trust Layer implements each of these:

Instruction Defense:
Prompts sent to LLMs include guidance instructing the model to exercise caution regarding what follows in the prompt.

Post-prompting:
Einstein products place user input near the beginning of the prompt where possible since LLMs often prioritize the instructions they encounter last.

Prompt Enclosure:
The Einstein Trust Layer isolates user-supplied input, preventing it from referencing other parts of the prompt by encasing user data by creating two separate prompts, with user data provided in one and other prompt information provided in another.

Length Restrictions:
This measure helps thwart DAN (Do Anything Now) style prompts and virtualization attacks by controlling the length of user input.

LLM02 Insecure Output Handling

Risk Overview

Insecure output handling vulnerability is a type of prompt injection that arises when a plug-in or application blindly accepts LLM responses without proper scrutiny and directly passes it to backend, privileged, or client-side functions. Since prompt input can control LLM-generated content, this behavior is akin to providing users indirect access to additional functionality.

Successful exploitation of an insecure output handling vulnerability can result in cross-site scripting (XSS) and cross-site request forgery (CSRF) in web browsers as well as server-side request forgery (SSRF), privilege escalation, or remote code execution on backend systems. The impact of this vulnerability increases when the application allows LLM content to perform actions above the intended user's privileges.

How Does Einstein Defend Against Insecure Output Handling?

The Einstein Trust Layer protects the Salesforce Platform against insecure responses generated by LLMs through a layered approach.

- Since the LLM is outside of the Einstein Trust Layer, the platform doesn't inherently trust any content generated by an externally hosted LLM. This approach treats externally hosted LLM-generated content with the same caution as any untrusted data source.
- Similar security controls that guard against prompt injections also shield the platform from insecure response handling. Strict prompt guardrails ensure the LLM generates responses that are consistent and aligned with system expectations.
- As part of their journey through the Einstein Trust Layer, responses from the LLM are subjected to meticulous screening and classification to detect toxicity and any irregularities.
- These generated responses then undergo conventional application sanitization measures to fortify defenses against potential XSS, CSRF, SSRF, privilege escalation, remote code execution, and agent hijacking attacks.
- After the generated data is screened, categorized, and sanitized, the Trust Layer transfers both the response and metadata derived from the screening process to the intended systems or components for subsequent processing.
- These interactions are all logged by systems in the Salesforce trust boundary and are used by our AI/ML Operations team to detect and address any anomalies. This data is kept for only 30 days.

Successful exploitation of an insecure output handling vulnerability can result in:

Cross-site scripting (XSS)

Cross-site request forgery (CSRF)

Server-side request forgery (SSRF)

Privilege escalation

Remote code execution

LLM03 Training Data Poisoning

Risk Overview

Training data poisoning occurs when an attacker or unaware client of the LLM manipulates the training data or fine-tuning procedures of an LLM. This ends up introducing vulnerabilities, backdoors, or biases that could compromise the model's security, effectiveness, or ethical behavior.

How Does Einstein Defend Against Training Data Poisoning?

Salesforce Trained Models
Salesforce is actively researching, developing, training, and providing several foundational models that are being used by both the Salesforce Einstein 1 Platform and the open-source community.

Salesforce research teams develop AI models using open-source datasets that are carefully vetted by Salesforce to meet Salesforce standards, legal obligations, and ethical objectives. Additionally, researchers can use anonymized and aggregated data in training these models, which is only done with explicit approval and agreement from customers whose data is included. Salesforce developers comply with industry standards including but not limited to OWASP 10, CWE25, NIST AI RMF.

The Salesforce AI Team helps identify and weed out poor-quality data sources with custom data cleansing and processing pipelines infused with artificial intelligence-powered tools.

Third-Party Models and Providers
When using third-party LLMs, the Einstein Trust Layer acts as a key protective mechanism in providing trusted LLM models. The Trust Layer uses Retrieval-Augmented Generation (RAG) to reference authoritative knowledge sources along with dynamic grounding which provides additional context to the user action by augmenting the prompts with instructions and information to prevent them from being factually inaccurate. This results in an interaction that is highly customized and appropriate to the customer based on their data.

While the usage of a diverse set of training data sources and robust model design makes it harder for attackers to manipulate the model by biasing it, the Salesforce security team uses a variety of process and technical measures to protect this data that may include but are not limited to:

- An in-depth review of our third-party LLM providers to ensure adequate controls are in place to counter the risk of malicious manipulation of training data.
- Internal security assessments to pinpoint and address vulnerabilities within third party large language (LLM) models. Salesforce conducts these reviews annually or upon any significant changes to the system design.

> Salesforce developers comply with industry standards including OWASP 10, CWE25, NIST AI RMF

LLM04 Model Denial of Service

Risk Overview

Model denial of service is a type of attack where an actor interacts with an LLM in a method that consumes an exceptionally high amount of resources. This results in a decline in quality service for them and other users, as well as incurring high resource costs.

How Does Einstein Defend Against Model Denial of Service?

Salesforce implements metering and rate limits in the Einstein Trust Layer as a first line of defense to prevent any single actor or application from causing service degradation.

Additionally, the Salesforce Secure Development Life Cycle has standardized and mandated the need for secure coding practices such as input validation and sanitization, which act as a further defense against denial-of-service attacks. In the current setup, all interaction with the Einstein Trust Layer requires user authentication and authorization.

A specialized AI/ML operations team continuously monitors all the Trust Layer activities through audit logs to ensure proper functionality. Any irregularities identified are thoroughly investigated and resolved to maintain the intended operation.

Hyperforce is the next-generation Salesforce infrastructure architecture, built for the public cloud.

The Einstein 1 Platform is hosted on **Salesforce's Hyperforce Environment**.

Hyperforce is the next-generation Salesforce infrastructure architecture, built for the public cloud. Hyperforce is composed of code rather than hardware, which allows the Salesforce Platform and its applications to be delivered rapidly. Built for resiliency and high availability, Salesforce Hyperforce environment uses three availability zones per region to locations worldwide, which provides Salesforce customers more choice and control over their data residency requirements. Salesforce products running on Hyperforce - including the Einstein 1 Platform, benefit from Hyperforce's integration of enhanced standards for compliance, security, agility, and scalability, and from Salesforce's continued commitment to privacy. Here's more information on the security of the Hyperforce platform. The Einstein 1 Platform Trust Layer abstracts the LLM provider from the Salesforce applications with a goal of providing choice and minimizing the impact of a failover.

Check out Hyperforce

LLM05 Supply Chain Vulnerabilities

Risk Overview

Supply chain vulnerabilities in LLM applications can affect the entire application lifecycle. This includes traditional third-party libraries/packages, docker containers, base images, and service suppliers such as application and model hosting companies. Vulnerable components or services can become the vector for cyber-security attacks leading to data disclosure and tampering, including ransomware or privilege escalation.

LLM applications that use their own models bring new types of vulnerabilities typically found in ML development. These include vulnerabilities in third-party data sets and pre-trained models for further training (transfer learning) or fine-tuning. Third-party data sets and pre-trained models can facilitate poisoning attacks, resulting in biased outcomes, security breaches, or complete system failures.

LLMs may depend on LLM plugins for extensions, which can bring their own vulnerabilities. LLM plugin vulnerabilities are covered in LLM Insecure Plugin Design, which covers writing an LLM plugin rather than using a third-party plugin. However, insecure plugin design provides information to evaluate third-party plugins.

How Does Einstein Defend Against Supply Chain Vulnerabilities?

Our products include third-party, open-source, and commercial components, such as libraries, images, and frameworks.

In addition to providing feature functionality, these components help engineers implement security controls such as input validation, output encoding, encryption, and web application vulnerability protection.

Before any such component is incorporated into our product, it must pass the security review process. This process starts with the engineering team working with the security team to assess the associated risks. These risks may span complexities, data classifications, new interfaces, third-party components, and the use of technologies like encryption and AI functionalities. As part of the Salesforce secure software development lifecycle, third party products are reviewed to ensure their compliance with industry standards.

Salesforce LLMs only use datasets that are carefully vetted by Salesforce to meet the rigorous standards, legal obligations, and ethical objectives set forth by Salesforce.

These components help engineers implement security controls such as:

- **Input validation**
- **Output encoding**
- **Encryption**
- **Web application vulnerability protection**

LLM06 Sensitive Information Disclosure

Risk Overview

Sensitive information disclosure occurs when an LLM accidentally reveals sensitive information, proprietary algorithms, or other confidential details through its responses. This can result in unauthorized access to sensitive data or intellectual property, privacy violations, and other security breaches.

How Does Einstein Defend Against Sensitive Information Disclosure?

The crux of this defense lies with the **Salesforce Platform Multitenant architecture** which allows multiple tenants to operate in isolation, for every tenant request, the platform kernel reads metadata and data from a shared database to provide each tenant with a customized application experience.

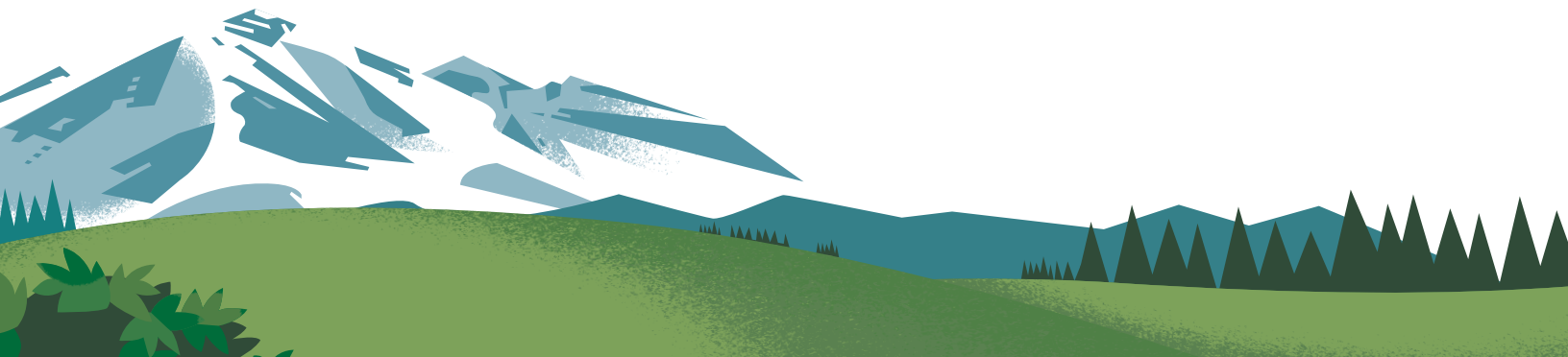
This design paradigm extends to the Einstein 1 Platform as well and ensures data isolation when interacting with an LLM.

Salesforce's development and data handling standards include strong guidelines around how developers and features use data sets that include sensitive data. Our standards require the removal or pseudonymization of data before it's used for training purposes.

Personal identifying information (PII) and Payment Card Industry (PCI) information require heightened controls to protect this sensitive information from potential exploitation. In the Einstein Trust Layer, data masking is used as a technique to obfuscate or remove specific elements of the prompt that identify individuals or reference sensitive or personal data.

Data masking is achieved by the replacement of sensitive data in a prompt with a masked version that can't be traced back to the original source before the prompt is sent to the LLM. Using Python Libraries and pre-trained models that provide named entity recognition, the PII in the prompts are identified and replaced with a token ("masking"). On the LLM generation, this token is then replaced into the original slots in the prompt before the final response is rendered ("demasking").

> Data masking is achieved by the replacement of sensitive data in a prompt with a masked version that can't be traced back to the original source before the prompt is sent to the LLM.



LLM07 Insecure Plugin Design

Risk Overview

LLM plugins are extensions that, when enabled, are called automatically by the model during user interactions. The model integration platform drives them, and the application may have no control over the execution, especially when another party hosts the model. Furthermore, plugins are likely to implement free-text inputs from the model with no validation or type-checking to deal with context-size limitations. This allows a potential attacker to construct a malicious request to the plugin, which could result in a wide range of undesired behaviors, up to and including remote code execution.

> Plugins are likely to implement free-text inputs with no validation, allowing a potential attacker to construct a malicious request, including remote code execution

How Does Einstein Defend Against Insecure Plugin Design?

On our hosted models, any need for a plugin will be subjected to the existing robust **Salesforce Secure Development Lifecycle** (SSDL) process.

Salesforce uses and follows an iterative security by design process, which is embedded in our policies, processes, and workflows. We created a system that's transparent and internally consistent through the implementation of security assessments, ongoing SSDL secure code training, threat modeling, ownership, and reflection. The result is a best-in-class process to address security throughout the software development lifecycle.

LLM08 Excessive Agency

Risk Overview

Excessive Agency is a vulnerability that allows damaging actions to be performed in response to unexpected/ambiguous responses from an LLM regardless of what is causing the LLM to malfunction. This includes hallucination/confabulation, direct/indirect prompt injection, malicious plugin, poorly-engineered benign prompts, or just a poorly-performing model. The root cause of excessive agency involves one or more of these cases: excessive functionality, excessive permissions, or excessive autonomy. This differs from insecure output handling, which is concerned with insufficient scrutiny of LLM responses.

LLM malfunctions include:

- Hallucination/confabulation
- Direct/indirect prompt injection
- Malicious plugin
- Poorly-engineered benign prompts
- Poorly-performing model

Root causes of excessive agency include:

- Excessive functionality
- Excessive permissions
- Excessive autonomy

How Does Einstein Defend Against Excessive Agency?

The Einstein Trust Layer builds guardrails by grounding the prompt or fine-tuning the foundation models with CRM data to ensure accuracy and relevancy.

AI workflows in our features take extra steps to ensure trust by including a human in the loop. This is exhibited in our generative AI features by enabling users to proofread, edit/update the model-generated texts before sending that content to end customers.

Moreover, the authorization for the generative AI components aligns with the application's underlying authorization models. Users without access to the underlying data in the application wouldn't be able to use generative AI features on that data.

A feedback framework captures the user's feedback, along with the unified feedback schema to capture the inputs and responses of Generative Services and LLMs.

LLM09 Overreliance

Risk Overview

Overreliance can occur when an LLM produces erroneous information and provides it authoritatively. While LLMs can produce creative and informative content, they can also generate content that's factually incorrect, inappropriate, or unsafe. This is called hallucination or confabulation. When people or systems trust this information without oversight or confirmation, it can result in a security breach, misinformation, miscommunication, legal issues, and reputational damage.

How Does Einstein Defend Against Overreliance?

Salesforce uses ML techniques such as regularization and adversarial training to ensure that our in-house developed models are constantly evaluated and improved upon.

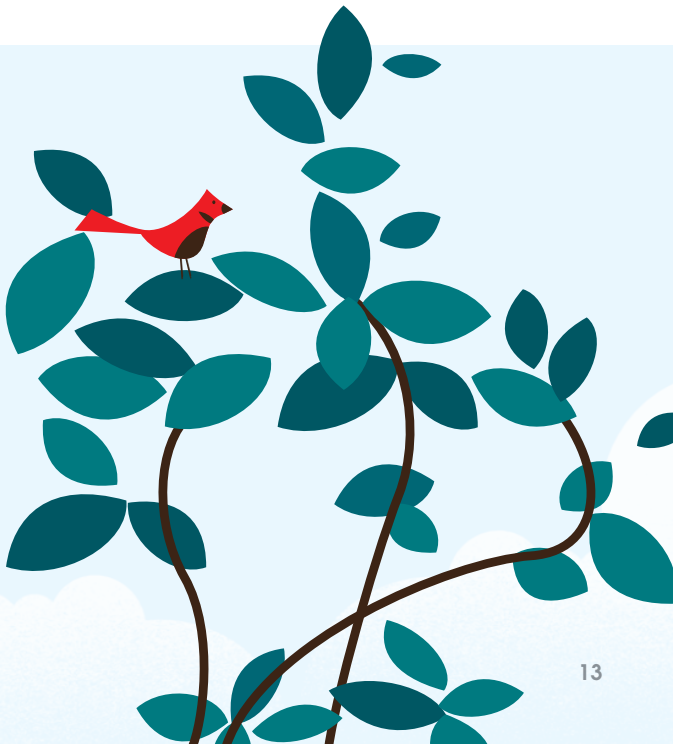
Salesforce uses prompt guardrails and grounding at multiple levels–including inside individual components and features and within the Trust Layer–to ensure LLM-generated responses align with Salesforce and customer expectations.

Salesforce also leverages "mindful friction" which ensures customers have the information they need to make the best decision for their use case. We help our customers stay ahead of the curve with unobtrusive but mindfully applied "friction" that interrupts the usual process of completing a task to encourage reflection. Examples include in-app design that provides guidance popups to educate users on possible bias or flagging detected toxicity and asking users to review the responses carefully before acting on them.

Additionally, to align with industry standards and our partners, and to protect our customers, Salesforce has published an externally-facing **AI Acceptable Use Policy**. This policy allows customers to use Salesforce products with confidence, knowing they and their end users are receiving a truly ethical AI experience from product development to deployment. The AI Acceptable Use Policy allows Salesforce to uphold the protections we promise customers, such as making sure that our products aren't causing avoidable harm. It also aligns us with third-party expectations when we work with partners opening up opportunities to collaborate in the market.

Salesforce's AI Acceptable Use Policy allows customers to use Salesforce products with confidence, knowing they and their end users are receiving a truly ethical AI experience from product development to deployment.

AI Acceptable Use Policy



LLM10 Model Theft

Risk Overview

Model theft arises when the proprietary LLMs are compromised, physically stolen, copied, or the weights and parameters used by the model are extracted to create a functional equivalent. Theft of LLMs represents a significant security concern as language models become increasingly powerful.

How Does Einstein Defend Against Model Theft?

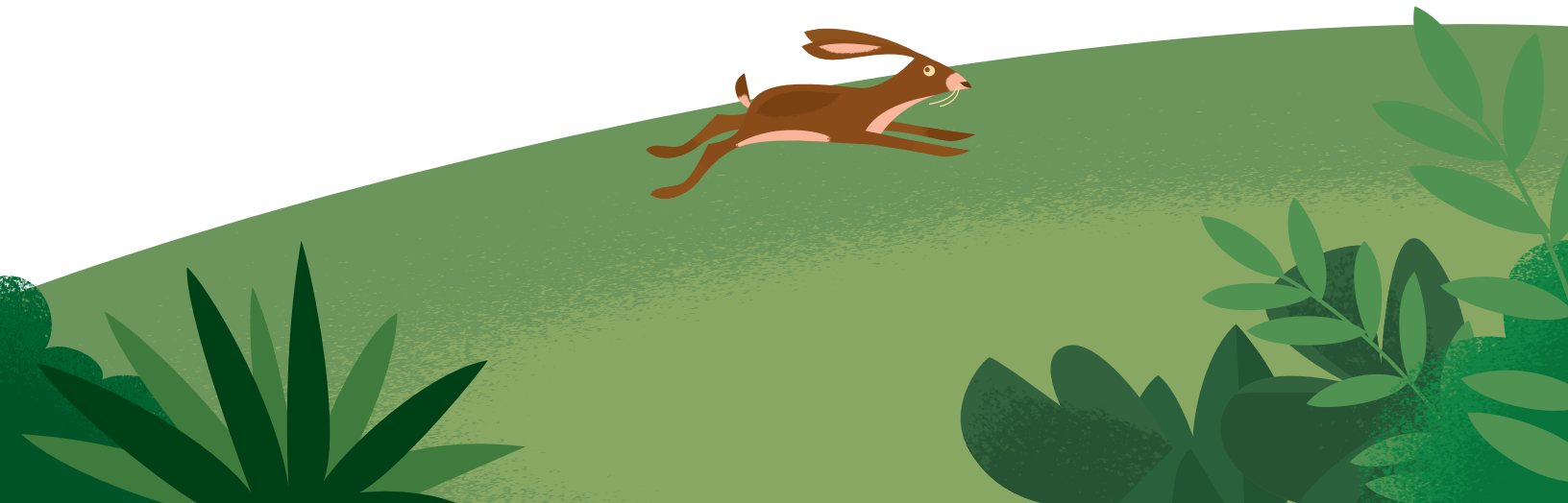
Access to Salesforce-hosted models is implemented in accordance with all other Salesforce production elements and conforms to existing security standards and best practices.

The controls used include requiring Just In Time (JIT) credentials, Multi-factor Authentication (MFA), strong audit trails, and logging.

Requests to the models used by Einstein are required to pass through the Einstein Trust Layer, which implements strong authentication and authorization controls through OAuth. This ensures that only authenticated and authorized clients can access the model.

The Einstein 1 Platform is hosted on **Salesforce's Hyperforce Environment**. Hyperforce is the next-generation Salesforce infrastructure architecture, built for the public cloud. It's composed of code rather than hardware, which allows the Salesforce Platform and its applications to be delivered rapidly and reliably to locations worldwide, giving customers more choice and control over data residency. Salesforce products running on Hyperforce benefit from its integration of enhanced standards for compliance, security, agility, and scalability, and from Salesforce's continued commitment to privacy. Here's more information on the **security of the Hyperforce platform**.

> Theft of LLMs represents a significant security concern as language models become increasingly powerful.



Conclusion

At Salesforce, we build security into every aspect of our business.

Process plays a critical part in maintaining customer trust, our number one value. Customers depend on us to safeguard their data, mitigate threats, and ensure our platform and services meet performance and security requirements. Our nimble approach to constantly evolving with the ever-changing threat landscape enables us to do exactly that.

"Security by Design" is more than just a catchy slogan for us; it is an iterative process embedded in our policies, processes, and workflow. We created a system that is transparent and internally consistent by implementing security assessments, ongoing training, threat modeling, ownership, and reflection. The result is a best-in-class process for addressing security at all levels of our application software development (from design to deployment).

If you have questions about our approach to handling security risks, please contact your account executive.