Low-Code_
Security_
Alliance

</>

SECURING SALESFORCE

Hard Truths about Shared Responsibility

BY THE SALESFORCE SECURITY EXPERTS OF THE LOW CODE SECURITY ALLIANCE:

Charan Akiri, Blanca León-Carter, Amit Chaudhury, John Crimmings, Andrew Davis, John Daniel, Rakesh Gupta, Gaurav Kheterpal, Joshua Kohl, Jason Lord, Matt Meyers, Matt Pieper, Jason Ross & Nicolas St-Pierre

Salesforce has empowered millions of trailblazers to build the applications that power the world's leading businesses. But most of these trailblazers incorrectly assume that security is Salesforce's responsibility alone.

You will not hear this message in Salesforce's marketing materials, but let us be clear: Securing Salesforce is your responsibility.

Those who understand the material in this guide have a unique advantage over those who naively assume Salesforce is inherently secure. Armed with this knowledge, you can protect your organization by setting clear and powerful standards for your teams and consulting partners.

The Low-Code Security Alliance is an independent group dedicated to ensuring a secure future for those building on low-code platforms like Salesforce. Membership is free and open to anyone who shares our mission. Join us at LowCodeSecurityAlliance.org

Salesforce, as one of the leading CRM platforms, has become integral to many organizations' operations. However, with its widespread adoption comes the challenge of ensuring robust security practices. This guide explores key problems in Salesforce development security, drawing insights from interviews with industry experts.

By 2026, low-code technology is expected to be used to build about 75% of all new applications.[1] For fifty years, the main constraint on the speed of development was the time it takes to write code. Low code systems make it faster and easier to develop. The explosion of AI is making development even faster and easier.

The biggest challenge in software development is gradually shifting from building new capabilities to ensuring that changes are safe. Organizations need to prepare today to ensure that secure and productive processes and habits are in place to avoid negative outcomes in the future.

Salesforce offers the ability to build low-code apps, but the security of these apps isn't guaranteed. The responsibility for security depends on the team building them. To properly prepare for the wide-ranging impacts of an increasingly low-code approach to application development, we need a deep understanding of where we are and where we are going.

THIS GUIDE EXPLORES SIX CRITICAL ASPECTS OF SECURING LOW-CODE DEVELOPMENT ON SALESFORCE:

1. The Current State of Salesforce Security
2. Common Security Vulnerabilities in Salesforce Development
3. The Risks of Security Failures on the Salesforce Platform
4. The Importance of Technical Leadership in Securing Salesforce
5. Obstacles to Educating Salesforce Developers on Security
6. Recommendations

EXECUTIVE SUMMARY

Salesforce is the leading enterprise business platform, having overtaken SAP in 2022. Salesforce's ability to keep companies on their platform has given them a gross revenue retention (GRR) of 92%[2] (8% attrition as of Jan 31, 2024) and a compounded annual growth rate (CAGR) of 12% in 2024. That growth has been largely fueled by enabling enterprises to build custom applications on top of their platform.

In Salesforce's first Annual Report as a public company in 2005, Salesforce said:

> *"We have redefined customization, making it easy for our customers to adapt our service to the realities of their businesses. That's a radical break from the standard enterprise software approach, which favors pre-built industry-focused solutions – called verticals – that assume all financial institutions, transportation companies, or manufacturers in the same or related industries are all alike. We call it 'the power to be unique,' and we believe that our reinvention of customization is a long-term differentiator for us. [This] ensures that Salesforce and custom applications written for our platform are integrated into the fabric of our customers' businesses."*[3]

While Salesforce customers vary dramatically in terms of their size and the degree of customization, for almost all large customers, Salesforce has long since ceased to be a simple SaaS solution. Salesforce has built a complete application development platform on top of their SaaS stack. They offer four custom programming languages (Apex, Visualforce, Lightning Web Components, and Aura), a custom visual programming language (Flow), and a huge range of other technologies for building custom applications.

Each of these technologies is proprietary to Salesforce and can be expressed as "metadata" to allow it to be stored in version control, deployed, or analyzed. In all, there are over 700 types of metadata. The proprietary nature of this metadata has two consequences:

1. It is extremely difficult to migrate this configuration to other platforms, meaning that those who build on Salesforce are locked-in to precisely the extent that they have customized the platform.

2. Generic security, compliance, and governance tools do not work on Salesforce, with limited exceptions.

True to their hopes in 2005, Salesforce has been "integrated into the fabric of our customers' businesses." But in doing so, these customers' businesses have been integrated into the fabric of Salesforce. In 2005 it was common for a company to have maintained 40 separate applications for different purposes. Over the last 20 years, many of those companies have consolidated those 40 applications into apps on the Salesforce platform. When they were separated, those applications each had their own security boundaries. But once they are consolidated onto the Salesforce platform, this becomes a single point of risk.

> **ANYONE WITH "VIEW ALL DATA" ACCESS TO THAT SINGLE DATABASE HAS "THE KEYS TO THE KINGDOM" AND COULD ACCESS ANY OF THE PROPRIETARY DATA STORED IN THAT SALESFORCE ORG.**

The data stored in Salesforce includes the crown jewels of personally identifiable information (PII): all information on a company's customers and constituents. More significantly, Salesforce Industry Clouds and the partner ecosystem have built applications that cast a wider net over the management of company data. This can include, among other things, contracts, inventory, credit card information, personal health information (PHI), and banking transactions.

The primary conclusion put forth in this guide is that a misunderstanding of Salesforce's Shared Responsibility Model is the leading factor causing organizations to ignore the very real issue of how to secure their Salesforce Orgs. Salesforce is secure by design, but that can quickly be undone when you start configuring it. To the degree that organizations customize Salesforce to their unique needs, its security falls into the Shared Responsibility Model. The Shared Responsibility Model means that customers are responsible for ensuring their Salesforce instance is configured and governed in a secure way.

In practice, Salesforce's primary value proposition is "time to value." They provide capabilities for low-code application development, work hard to equip millions of developers (with a wide variety of experience), and partner with armies of third-party system integrators or contractors to ensure that customers can build applications as quickly as possible.

But for the teams focused on getting Salesforce applications released quickly, secure configuration is rarely a significant consideration. Popular career guides for Salesforce mention many roles but omit Salesforce Security Specialist.[4] In the unusual cases where security is considered in the development process, it is often an afterthought and

limited to only subsets of the application.
The underlying assumption is that Salesforce is secure. But people systematically fail to distinguish between the security of the underlying platform, and the security of the customizations built on top of it.

It is unknown just how many Salesforce sites are leaking data, but researchers have confirmed at least hundreds[5], primarily through Salesforce Digital Experience sites. But the vulnerabilities are not limited to that. In all but the most trivial of cases, the complexity of Salesforce has grown beyond what individual humans can understand. If you cannot understand the Org, you cannot secure it.

As a company, Salesforce must balance how they talk about securing customer configurations on the platform. The more they draw attention to the risks of insecure configurations, the more they undermine their core value proposition that the platform is inherently secure. This makes it even more important for customers to understand how they can ensure security.

01 THE CURRENT STATE OF SALESFORCE SECURITY

Salesforce—as a platform—provides a robust foundation for application-level security. While this is critically important, it is easy for users and those who manage a Salesforce Org to confuse the security of the underlying platform with the security of the data model built on top of it. This leads to a false sense of security. For example, Salesforce ensures that no one is accessing its underlying servers and networks. But the data that customers store in Salesforce is protected or left vulnerable depending on the way Salesforce is configured by customer teams.

Understanding the many layers of Salesforce's data access model requires careful study. Salesforce offers system, object, field, and record-level access and security controls, in addition to multifactor authentication, or MFA, and paid add-ons like Shield and Security Center. Understanding how this data model plays out in a particular Org requires detailed analysis. The burden of identifying and implementing security architecture within an Org has historically been an assumed responsibility of Salesforce Admins. But offering security controls alone does not guarantee they will be properly architected or even used at all because of the complexity of considerations and the variety of needs across organizations.

The most attractive feature of Salesforce is what makes it most dangerous. Being a low-code platform makes it faster and easier to build applications compared to traditional development. Yet, the more quickly a platform can change, the more quickly it can become insecure. The heightened rate of innovation on Salesforce brings with it greater risk.

As Salesforce Orgs inevitably evolve after the initial implementation, applications that were thoroughly secure one moment can quickly become insecure with just a few clicks from an unsuspecting Admin. Therefore, it is imperative that security is woven into a team's ongoing operational processes and augmented with automated tools.

Salesforce provides its own free Security Health Check app[6] that can identify some vulnerabilities. They also provide tools like Shield and Security Center, although these can cost as much as 30% of the base Salesforce license price. Shield capabilities such as event monitoring are standard free capabilities on some other platforms. Salesforce is clearly aware of the risks to those who build on their platform, but those concerns are forever balanced against the opportunity to increase revenue.

Each organization's security posture is different. The evaluation of risk is something that each organization must take upon themselves.

1.1. HOW WE GOT HERE

Customers have invested phenomenal amounts of money in Salesforce—$30 billion a year in annual revenue. And that's often millions or tens of millions of dollars per year per customer.

The primary goals in purchasing Salesforce are generally to get basic SaaS functionality on a hyper-secure enterprise platform where Salesforce takes care of all the infrastructure. On top of this, the Salesforce platform allows companies to customize the SaaS applications or build custom business applications with relative ease.

Most Salesforce implementations are initially done through consulting partners. "Nine out of 10 customers rely on our partner apps and experts, and 70% of Salesforce implementations are led by one of 132,000 credentialed Salesforce experts."[7]These initial Salesforce implementations are run as projects, focused on one-time delivery rather than ongoing maintenance. Consulting partners hustle to deliver on time and on budget. The rush to build often leaves Orgs with a tangled web of dependencies and components, and because the implementation partners eventually leave, there is a risk that institutional knowledge will leave with them. Even when consulting partners offer managed services for an Org, the churn of consultants is typically higher than churn for full-time employees.

Each Salesforce Org is a single, tightly coupled monolithic database. As the configurations and custom solutions of an Org grow, so does the complexity of ensuring changes are secure. When security is not integral to the architecture and development of a Salesforce Org, the cost and difficulty of securing it retrospectively becomes far higher, making it difficult to make deep structural changes.

Salesforce was not originally designed to be a development platform. It was designed 25 years ago to be a CRM. But the underlying monolithic structure of a Salesforce Org has become the foundation for millions of apps serving 150,000 businesses. The cost to change that underlying architecture is too high. The only option is to invest in securing that platform to protect an organization's most sensitive asset: customer data.

1.2. TAKING THE SHARED RESPONSIBILITY MODEL SERIOUSLY

As with other cloud providers, Salesforce employs a *Shared Responsibility Model* for security on their platform. This means that Salesforce secures the underlying infrastructure and makes it **possible** to secure your Salesforce instance. While there are

opportunities for Salesforce to provide more guidance for their customers, the security of each specific Salesforce Org is the customer's responsibility.

When organizations work on Salesforce, they often assume the underlying security of the Salesforce platform implies security of their instance. This attitude is also promoted by some Salesforce AppExchange partners, who claim that because their apps are built on Salesforce, they are inherently secure. But every Salesforce Org is uniquely customized for a particular company, and the challenge of ensuring security multiplies with every additional component.

The shared aspect of security deserves more attention than it gets. For example, Salesforce's "Salesforce Security: Commitment to Customer Trust" document is over 20 pages long, detailing the steps they take to secure their underlying infrastructure. It is not until the last page that the Shared Responsibility Model gets introduced:

> *"When you become a Salesforce customer, you gain a trusted digital advisor and partner in protecting your data. As part of our commitment to safeguarding customers' data, Salesforce:*
>
> - *Provides a robust security-focused services infrastructure*
> - *Enables customers to operate their Salesforce environments securely by providing tools, technologies, professional services, and sales engineering support*
> - *Educates customers on the need and options for enhanced security features*
>
> *"In turn, we strongly urge our customers to:*
>
> - *Monitor user behaviors and event logs in their environments*
> - *Protect sensitive customer data in alignment with compliance standards*
> - *Adopt the latest, available security controls and features"*

Our experience has shown that while these last three responsibilities may seem straightforward and based on common sense, implementing, and scaling them across an entire Org can be extremely difficult without specialized tooling or expertise.

Leaders must be aware of these shared responsibilities and not just assume risks are addressed. Awareness, though, is only half the battle. They must also communicate the severe consequences of vulnerabilities and data breaches to their teams and ensure teams are well-trained and equipped with appropriate tools and knowledge. Failure to

do so opens the possibility for security gaps to arise.

1.3. SALESFORCE'S SLOW MARCH TOWARDS SECURE-BY-DEFAULT

Salesforce is gradually changing default behaviors to reduce the risk of companies leaking data, especially on Digital Experience Sites.[8] But organizations that configured their Orgs prior to these new, more-secure defaults are more likely to have insecure configurations.

A relatively early restriction that Salesforce rolled out in 2021 rescinded the "View All" permission for Guest users.[9] This restriction did not automatically apply to prior customizations, and many loopholes exist that open the possibility of Guest users inappropriately viewing all records. For example, Apex code runs in System mode by default, which could potentially expose data externally.[10] Flows are similarly insecure unless carefully configured.[11]

Another example relates to external user security. Prior to 2020, the Organization-Wide Defaults for sharing data to external users, like non-employees who interact with an organization's CRM data, was set by default to the same level as sharing for internal users.[12] While this has since been changed to make external sharing default to Private, it does not automatically restrict access that was configured before 2020.

Why doesn't Salesforce enforce these security updates for all customers automatically? Some of these security updates have the potential to break perfectly safe and intentional customer applications. Salesforce cannot know with certainty if your implementation is designed securely to fit your use case. They must strike a balance between raising the bar on security in a way that does not break customer applications.

1.4. CALCULATING YOUR SALESFORCE ATTACK SURFACE

The Open Worldwide Application Security Project (OWASP), a leading IT security standards body, outlines how to analyze the Attack Surface of applications like Salesforce:[13]

> *"The Attack Surface describes all of the different points where an attacker could get into a system, and where they could get data out.*
>
> *The Attack Surface of an application is:*
>
> 1. the sum of all paths for data/commands into and out of the application, and

2. the code that protects these paths (including resource connection and authentication, authorization, activity logging, data validation and encoding)

3. all valuable data used in the application, including secrets and keys, intellectual property, critical business data, personal data and PII, and

4. the code that protects these data (including encryption and checksums, access auditing, and data integrity and operational security controls).

You overlay this model with the different types of users - roles, privilege levels - that can access the system (whether authorized or not). Complexity increases with the number of different types of users. It is important to focus on the two extremes: unauthenticated, anonymous users and highly privileged admin users (e.g., database administrators, System Administrators).

Group each type of attack point into buckets based on risk (external-facing or internal-facing), purpose, implementation, design and technology. Then, count the number of attack points of each type. Next, choose some cases for each type. Finally, focus your review/assessment on those cases."

This provides a rough approximation of how to calculate the risks. Note that complexity increases with each new part of the application, and "complexity increases with the number of different types of users." Organizations fail to appreciate that each new piece of Salesforce configuration increases your attack surface. The larger the attack surface, the greater the risk of a breach, and the greater the cost of ensuring security across the application.

Just like people accumulate credit card debt in the rush to acquire new possessions and experiences, organizations accumulate security debt in the rush to configure Salesforce.

02 COMMON SECURITY VULNERABILITIES IN SALESFORCE DEVELOPMENT

The first step toward protecting your system against security vulnerabilities is to learn about them. Here are four major vulnerabilities we have seen across numerous Salesforce instances:

2.1. EXCESSIVE PERMISSIONS

Excessive permissions continue to be one of the most serious vulnerabilities facing Salesforce users. Every additional user who can access a piece of data increases the potential for accidental corruption and exposure. Minimizing access to data based on personas—especially where Salesforce is used in high-risk environments like the public sector, healthcare, and finance—needs to be a priority.

Excessive permissions typically result from lacking a security architecture defining what personas within an Org should be able to see or do. The lack of clear architecture can lead to unauthorized access of sensitive data and introduce the risk of breaches and industrial espionage. The absence of a documented security architecture also makes change management difficult, which can exacerbate these issues.

Having countless combinations and permutations of Salesforce access controls creates an environment where insecure permissions can multiply and hide. Permissions accumulate layer upon layer the more development happens in an Org. Profiles, the main Salesforce access controls, are a bit of a black box. The Profile UI in Salesforce does not allow Admins to easily see the full list of permissions granted to a specific Profile. Retrieving Profile metadata as XML is similarly challenging. This makes it hard to fully understand what permissions a profile has been given. Permission Sets and Permission Set Groups are a less opaque alternative that are gradually becoming the main access controls. But Permission Sets too must be kept as simple as possible to reduce the surface area for vulnerabilities to hide inside.

These issues perpetuate another problem – overly-permissive Profile assignments. For example, Profiles for System Administrator permissions are often assigned too liberally. Integrations are also frequently configured using System Administrator privileges, instead of a more restrictive Integration User profile. Confusion about what permissions a Profile grants, combined with liberal Profile assignments, compound security risks. Salesforce also combines multiple elevated system permissions together under a single setting, making it difficult to define a least-privileges model for System Administrators. Inevitably, more users have System Administrator privileges compared to an ideal strategy to limit access.

How to Address These Issues

Understanding the personas of users in an Org and defining and documenting what they should be able to see and do are the best ways to address the issue of excessive permissions. This security architecture becomes the North Star for maintaining and updating permissions.

Defining and documenting security for the various personas within an Org helps mitigate the problem of having too many Profiles or Permission Sets as well. When you know exactly what a type of user should be able to do or see, it is easier to identify superfluous or redundant Profiles and Permission Sets. This is important because it reduces the security overhead to maintain and scale an Org. For example, large numbers of Profiles and Permission Sets make it difficult to debug security issues—the more redundant layers of security, the more time spent identifying the root cause of a problem.

Organization-wide defaults (OWD) are also key to defining a strong security architecture. Start with the principle of least privilege and set OWD to *private* by default. This *Secure-by-Default* configuration will be the foundation of the persona-driven security model of permissions by reducing opportunities for over-permissioning.

Limit users with long-term elevated permissions and ensure these permissions are only invoked when there is a valid business reason to have them. Build auditable mechanisms to quickly assign and remove elevated system permissions as needed to meet the unique needs of the Salesforce Org.

Finally, proper training and awareness among administrators, developers, and business leaders about the risks of excessive permissions are essential. Leverage a well-documented persona-driven security architecture to drive training and awareness and serve as a roadmap for how new permissions are assigned. Org security requires continuous attention to verify proper adherence to organizational and regulatory requirements for data security. A persona-driven security architecture will make recommended regular audits easier to perform, which makes detecting and mitigating issues simpler. This result is a cleaner, more secure Org, unburdened by excessive permissions.

2.2. INSECURE SITES, COMMUNITIES, AND DIGITAL EXPERIENCES

Salesforce has long offered solutions to provide external users customized *portal* access to the underlying Salesforce instance to provide web forms and completely customized user experiences. While we are using the term “Community” to discuss these risks, these technologies are known by many names, including Portals, Sites, Communities, and Digital Experiences. It is important to understand that a strong security architecture pays equal attention to both internal and external users of an Org. Nowhere is this more important—and nowhere else have the consequences been so severe—than with the configuration of these features. Community security is controlled by guest user and external user permissions. When not properly configured, these configurations can lead to unintentional exposure of sensitive data, which can result in massive disruptions, exposure to damages, and threats to regulatory compliance.

Communities require an extra layer of security administration and configuration. Without proper training and awareness, it can be easy to misconfigure a Community’s security, exposing your sensitive data to the world. Most people do not realize that Communities are powered under the hood by client-side facing APIs. These APIs are used to render the page layouts, the fields on the page, and most importantly, retrieve and display the data on the page. This means that even if your sensitive data is not displayed on the page, it could still be accessible via these APIs. It’s even more critical regarding Guest Users. If not properly configured in your security architecture and sharing model, anyone could access these APIs and your data without even authenticating. For example, if you created a criteria-based sharing rule granting read-only access to the Guest User for access to Cases, you just exposed all your Case record data to everyone on the Internet.

How to Address These Issues

Organizations must take a hard look at Community configurations to avoid accidental exposure and ensure proper access controls. Communities are easy to launch but take time to secure.

Addressing insecure Community configurations involves setting robust access controls and ensuring that Community permissions are tightly managed to prevent data leakage. Here are a set of specific recommendations:

- Communities add layers of complexity to the existing internal data model. If your default data access model (record-level sharing and FLS) is not well-defined, start by improving your data model for internal personas. Communities often extend the existing user interfaces, and a well-defined internal data access model adds a critical layer of protection from easy misconfigurations and coding errors.

- Many Communities introduce new authentication methods to your Salesforce org. Multifactor authentication, or MFA, and other authentication best practices are applicable to your Community launch. Only use secure authentication methods or a combination of methods; usernames and passwords alone are rarely enough.

- If all users are authenticated and anonymous access is not required, disable the guest user by removing all checkboxes on the associated guest profile and adding login restrictions. If you do need some level of anonymous access, remove all system access settings from the guest user profile and only add back what you understand and what is required for your community to function.

- Set the object-level external organization-wide defaults to private and only open data access as needed for users to meet their objectives.

- In September 2019, Salesforce removed the ability for Guest Users to own records, but this change only applied to new records. We recommend auditing all records prior to the change to ensure that a Guest User is not the owner of any records, putting your data at risk.

- Review all your Criteria-Based Sharing rules to ensure that no sensitive data is shared with a Guest User. Only share records with a Guest User that you are okay with exposing to the entire world.

- Disable the setting that allows Guest Users to upload files to Salesforce. Files pose a substantial risk to your data, as they can contain viruses and other malicious threats. Allowing files to be uploaded by a Guest User opens your organization up for attack, as anyone could upload malicious files to your Salesforce environment since Salesforce is not scanning these files for viruses or other threats.

- Disable the setting to "Allow the Guest User to See Other Members of This Site." With this setting enabled, Guest Users have access to view any user in the Community.

- Enable the "Show Nicknames" setting to hide Community user's first and last names. Instead, only the nickname for the users will be displayed.

- Insecure custom code can rapidly scale a vulnerability in Communities. Implement code reviews and code analysis on any code customizing Communities.

- Audit your Community regularly. Communities are easy to quickly launch and modify. Regular audits of guest users and Community settings are critical to maintain a secure Community. External audits are recommended

2.3. INSECURE APIS

APIs are a critical part of Salesforce integrations, but they can also be a hidden source of security vulnerabilities.

Potential issues include:

- There are at least 140 Salesforce API endpoints, many of them not listed in Salesforce documentation. These API endpoints are used to enable the functioning of Salesforce Digital Experiences and other capabilities. If data is not secured, some of these APIs can be queried to extract data.

- Platform events where the source is not checked, and payloads are taken as-is.

- Interactions with endpoints and how they are called, as calling an endpoint does not guarantee its security.

- Instances where the integration user was given System Administrator privileges, greatly increasing the risk of other insecure API and development practices.

- Limited options to adjust the scope on Connected Apps, contributing to the insecurity of APIs.

- Potential for token leakage through insecure APIs.[14] APIs should almost never allow unauthenticated access. Tokens are the equivalent of a password that is included in an API request to indicate that the request is authentic. But just like passwords, if tokens are not secured, data can be accessed inappropriately.

How to Address These Issues

Every integration with your Salesforce environment needs to be continuously monitored. The settings and configurations need to be investigated and maintained with frequent audits and verification for adherence to internal standards and policies.

- Approach each integration with a zero-trust mindset and apply system-level and data access permissions only as necessary.

- Do not share integration users across multiple integrations. This is even more critical when granting third parties access to your Salesforce data.

- Use strong authentication for all integrations.

    - Salesforce Named Credentials should use the strongest possible authentication type for the systems they access.

    - Connected Apps with OAuth should be used for all inbound integrations.

    - Never use password authentication for integrations.

- Enforce IP restrictions for all integrations.

- Remove unplanned API access. Disable API access from all users by default and only allow secure access to APIs through Connected Apps when there is a business need to do so. Remove or disable unnecessary Connected Apps.

- Build a strong data access model and do not use page layouts to hide data or prevent data from being manipulated. Data that is hidden or read-only on a page layout can still be visible or editable via the API.

2.4. CODE-LEVEL VULNERABILITIES

A proactive approach to security by developers is important to prevent vulnerabilities that result from improper coding. Manual code reviews by a skilled reviewer knowledgeable about security best practices are important to ensure secure development. However, these processes are time-consuming and prone to human error.

Custom Apex code can introduce security vulnerabilities. Apex classes should generally be defined "with sharing" to ensure that record-level sharing is enforced. SOQL queries should generally be defined "WITH USER_MODE" to respect object-level, field-level, and record-level sharing. Rushed or careless development is often the culprit for these insecurities, but the underlying cause is typically a lack of curiosity and proactive security measures by development teams.

The rush to populate the Salesforce world with trained developers has also led to a rash of "fake seniority," where individuals are assumed to be more knowledgeable than they are. Senior developers should be expected to write secure code. But it is common in the Salesforce world to mistake Certifications or tenure for deep knowledge.[15] It is not uncommoncommon for highly certified and long-tenured Salesforce specialists to lack the knowledge and experience necessary to create reliably secure code.

Just as we could incorrectly assume that Salesforce developers write secure code, we can incorrectly assume that LLMs like GitHub Copilot, ChatGPT, and EinsteinGPT write secure code. The security standards body OWASP warns against "excessive reliance on LLMs"[16]:

> *"While LLMs can produce creative and informative content, they can also generate content that is factually incorrect, inappropriate or unsafe… 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."*

It also happens that developers occasionally circumvent security measures to solve immediate problems, such as disabling encryption to address a particular bug, while ignoring the underlying risk.

How to Address These Issues

Static code analysis tools are critical for identifying and fixing code-level security issues. These automated tools go a long way to provide the support developers need to create safe code. This is especially true with linting tools embedded in the IDE, which provide ongoing feedback that can address knowledge gaps of developers. However, developers must be trained with an emphasis on secure coding practices to mitigate code-level vulnerabilities. Security analysis tools can catch the more obvious security issues, allowing manual reviews to identify more subtle security flaws.

The emerging LLM tools can often provide insights into and remedies for vulnerabilities. But teams should be careful to “trust but verify” that the remedies themselves are secure.

Comprehensive DevSecOps tools that include static analysis and CI/CD testing capabilities expand an organization’s capacity to monitor and manage code security by integrating directly into the development process.

03 THE RISKS OF SECURITY FAILURES ON THE SALESFORCE PLATFORM

What are the risks of allowing insecure configurations on Salesforce?

3.1. LOSS/CORRUPTION OF CUSTOMER DATA

Data loss or corruption is a major risk. Proper attention to the setup and maintenance of internal processes is critical to avoiding damaging consequences. Here are some of the main factors that lead to costly data loss or corruption:

- **Inadequate Data Backup Solutions:** Companies often incorrectly assume that Salesforce's data redundancies imply there is no need for external data backups. Such misunderstandings cause companies to slash their budget for data protection, putting it at risk for data loss or corruption. Salesforce is very unlikely to lose customer data, but customers are highly likely to need a way to reverse self-inflicted data loss.

- **Insider Threats:** Insider threats are a significant risk factor contributing to the loss or corruption of customer data.

- **Poor Change Management:** Lack of rigor around change management can cause data corruption. This includes Admins with excessive permissions in production who can inadvertently cause issues.

- **Importing Data:** It is easy for data to be corrupted during the importing processes. Robust backup tools and disaster recovery (DR) plans are essential, and it's critical that staff know how to respond to data loss incidents.

- **Exporting Data:** Without careful controls, it is possible for Salesforce users to export data inappropriately. For example, it can be tempting for a salesperson to export lists of accounts, contacts, and opportunities when they leave a company.

- **Misunderstanding the Shared Responsibility Model:** Companies can incorrectly assume that Salesforce's robust infrastructure implies they do not have to worry about data loss. Overreliance on Salesforce leads to a lack of preparation for data loss incidents.

- **Third-Party Integrations:** Issues arise when third-party integrations require System Administrator privileges or *Modify All Data* permissions, significantly increasing the risk of data loss.

- **Ease of Data Corruption:** Tools like Data Loader that make bulk data changes on Salesforce are widely accessible, making data corruption easy and common.

Robust data protection practices are essential to mitigate this risk.

3.2. INDUSTRIAL ESPIONAGE AND COMPETITIVE LEAKS

The potential for industrial espionage and competitive leaks is a serious threat, particularly in high-stakes environments. Salesforce is the system of record for proprietary customer and partner deals, contracts, M&A information, drug trial data, banking data, healthcare, and an increasing amount of other sensitive information. Protecting sensitive data from unauthorized access is crucial.

Global corporations can have a strategic importance to national security and power. Industrial espionage of U.S. companies by foreign governments has become so common that the White House listed threats from foreign governments as a main source of concern in its March 2023 cybersecurity strategy.[17]

These types of threats do not stop there, though. Competitive leaks from industry insiders are also a common source of data exposure. When it comes to business, organizations are always trying to find an edge against their competition. It does not take more than one bad actor to expose company secrets and give competing businesses the insight they need to take control of the market.

3.3. REPUTATIONAL DAMAGE AND REGULATORY FINES

Security breaches can lead to significant reputational damage and regulatory fines. The long-term impact on trust within an organization and the difficulty in regaining confidence once lost are considerable.

Data breaches significantly increase the risk of losing customers. Lost customer revenue hits twice: the loss of revenue and the cost of acquiring new customers to replace the ones who fled. Imagine if a large enterprise lost even 10% of their customers. That could account for millions or even hundreds of millions in losses.

IBM's Cost of a Data Breach report[18] describes the average total cost of a breach:

> *"The average cost of a data breach jumped to USD 4.88 million from USD 4.45 million in 2023, a 10% spike and the highest increase since the pandemic. A rise in the cost of lost business, including operational downtime and lost customers, and the cost of post-breach responses, such as staffing customer service help desks and paying higher regulatory fines, drove this increase. Taken together, these costs totaled USD 2.8 million, the highest combined amount for lost business and post-breach activities over the past 6 years."*

A breach puts the focus on the affected team. Everyone—from management to impacted customers—will want to know why the leak occurred, who let it happen, and what will be done to ensure it does not happen again.

Breaking this trust can never be fully repaired. Every team says they are secure until something bad happens. Why would users trust a company if they were wrong about being secure in the first place?

Another consideration for the impacts of data loss events is the potential for regulatory fines and penalties. Industries like healthcare and finance, or any sector handling sensitive information, are liable to incur these penalties if they cannot maintain control over who is accessing their sensitive information.

04 THE IMPORTANCE OF TECHNICAL LEADERSHIP IN SECURING SALESFORCE

Ultimate accountability for securing Salesforce rests with the executive leaders who oversee their Salesforce implementations. These leaders often bear many responsibilities that can be difficult or overwhelming to balance. It is imperative that they find a path to securing the enterprise that is gradual and sustainable.

Shared Responsibility does not mean leaders need to understand every nuance of Salesforce security. But it means they should take a skeptical and thoughtful approach to ensuring each member of their teams maintains the highest standards. This means that the only scalable way for senior leaders to ensure security on Salesforce is to build up the security intelligence of every person on their teams.

It is the leader's responsibility to set clear expectations for the quality, priority, and security of work performed by all employees and contractors under their guidance. It is common for organizations to simply assume that work done by third-party system integrators or contractors is secure. But this standard must be inspected to ensure it's upheld.

Security risks are complex by nature; they change continually, and every security intervention leads to a response by the organization. If security protocols are too strict, employees may bypass them to get their work done. If they are too lax, employees may expose the entire organization without realizing it.

Managing a complex risk like securing Salesforce requires a scientific thinking approach. The preeminent approach to building scientific thinking[19] in organizations is Lean, also known as the Thinking People System. And the preeminent approach to improving team-level performance[20] and the ability to change[21] is transformational leadership.

Transformational leadership is essential for enhancing organizational performance, particularly within DevOps and software delivery contexts. Research by Google's DevOps Research and Assessment group (DORA) shows that effective transformational leaders drive high performance[22] by fostering an environment of innovation, continuous improvement, and personal growth.

Transformational leadership emphasizes key behaviors that significantly impact

technical and product management practices, ultimately improving Salesforce development security.

4.1. KEY BEHAVIORS OF TRANSFORMATIONAL LEADERS

Transformation leaders need to be flexible to the unique needs of their organizations, but there are key characteristics that effective leaders share:

1. **Vision:** Leaders provide a clear and compelling vision for the future, helping teams understand long-term goals and the path to achieving them.

2. **Inspirational Communication:** Leaders use positive reinforcement and motivational communication to instill pride and enthusiasm in their teams.

3. **Intellectual Stimulation:** Leaders challenge their teams to think critically and creatively, encouraging them to question assumptions and explore new innovative solutions.

4. **Supportive Leadership:** Leaders show genuine concern for the well-being and personal development of their team members, fostering a supportive and inclusive environment.

5. **Personal Recognition:** Leaders acknowledge and celebrate the achievements and improvements of their team members, reinforcing positive behaviors and boosting morale.

4.2. IMPACT ON SALESFORCE DEVELOPMENT SECURITY AND ORGANIZATIONAL PERFORMANCE

Transformational leadership indirectly influences organizational performance by enabling the adoption of key technical and product management practices.

The behaviors of transformational leaders drive the implementation of these practices, which strengthen Salesforce development security and improve organizational outcomes.

Ensuring security requires deep commitment from the entire development team to the long-term health and safety of the Org. The importance of security can easily be overlooked, unless the team incorporates that thinking into every activity. It is the responsibility of leaders to drive such thinking.

4.3. ADDRESSING LEADERSHIP GAPS

The responsibility of leaders is to align an organization's values, bring clarity around the challenges and mission, and promote the flow of work and value throughout the organization. Transformational leaders align their organization's vision with actionable goals, ensuring that everyone from top management to individual developers understands their role in achieving these goals.

There is a natural gap between the priorities of individual developers, organizational leaders, Salesforce, and those who entrust companies with their data, but poor data security is a critical risk shared across all parties. It is leaders' responsibility to bridge the gap in natural inclinations and to effectively communicate the importance of top-down security practices.

It is counterproductive to place blame on any single company, team, or team member. The work of all parties is interdependent. The meaning of Shared Responsibility is that each individual and organization must transcend their narrow concerns to address this broader challenge. Individuals must learn to strengthen security practices through proper training and guidance. Maintaining safe and clear lines of communication between individual contributors and leaders is necessary for everyone to feel comfortable talking about these issues. This also empowers individual stakeholders to further these security initiatives.

4.4. ADDRESSING LEADERSHIP GAPS

Transformational leaders cultivate a culture of continuous improvement, collaboration, and shared ownership, sustaining long-term success.

The benefits of strong leadership include:

- **Enhanced Organizational Performance:** Effective leadership drives the adoption of proven practices, resulting in higher efficiency, quality, and customer satisfaction.
- **Improved Salesforce Development Security:** High-performing teams consistently exhibit strong transformational leadership behaviors, leading to better security outcomes.
- **Reduced Technical Debt:** Transformational leaders champion a culture of care, ensuring that work is performed to a high standard. An example of this is always leaving the codebase cleaner than you found it, considering the impact on future developers and users.

4.5. ENABLING VALUE, CLARITY, AND FLOW

Transformational leaders ensure they are consistently improving their own sense of value, clarity, and flow while helping the rest of the organization do the same.

Understanding the broad implications of security helps team members understand their place in the larger strategy. The team must appreciate the risks and benefits associated with various activities. This includes developers understanding security risks and the importance of protective measures, empathizing with at-risk customers, and recognizing business risks.

Maintaining the integrity of information passed throughout the organization is important. Much like in the game of telephone, information can be diluted or altered as it travels through an organization. It is the leader's responsibility to address this distortion of information and preserve the original vision by maintaining clear and consistent communication.

Consistently clear communication ensures cohesive motivation across the organization. Transformational leaders work to synchronize efforts, simplify processes, and modularize tasks to enhance coordination and efficiency.

4.6. ADVICE FOR IMPLEMENTING TRANSFORMATIONAL LEADERSHIP

Starting down the path toward transformational leadership can seem daunting, but a gradual and thoughtful approach will produce results in a surprisingly short time.

The first step is to redefine leadership as a collective action, moving towards a collective goal. Leaders and followers are mutually dependent and come into existence together. Leadership is a social activity, not an individual activity.

> *"If you think you're leading, but no one is following, then you are only taking a walk." - John Maxwell*

From there, keep these five considerations in mind as you begin reassessing current practices and looking for potential improvements:

- **Set Clear and Measurable Goals:** Establish specific, achievable objectives for your team that align with the organization's vision.
- **Encourage Open Communication:** Foster an environment where team members feel comfortable sharing ideas and feedback. People's willingness

to raise bad news is a direct measure of the level of psychological safety in the organization. People bring their personal and cultural tendencies with them to work. But leaders must systematically reduce the psychological risk people feel in sharing information.

- **Promote Continuous Learning:** Provide opportunities for professional development and encourage team members to expand their skills.

- **Celebrate Team Accomplishments:** Regularly acknowledge the efforts and achievements of the team to boost motivation and morale. Cultivate an atmosphere of rejoicing in collective success. Be careful to avoid falling into the common trap of using praise and criticism as rewards or punishments. Praise and criticism are two sides of the same coin, and they can both reinforce a patronizing dynamic where leaders retain most of the power. Using praise and criticism to drive behavior does not drive the intrinsic motivation that is critical for learning organizations.

- **Lead by Example:** Demonstrate the behaviors and values you wish to see in your team, serving as a role model for others to follow. As often as possible, roll up your sleeves and work alongside the team in resolving issues.

By embracing these principles and practices, transformational leaders can significantly enhance their team's performance and contribute to the overall success of their organization. Transformational leadership is not only about guiding the team through change but also about creating an environment where every team member can thrive and contribute to the organization's long-term goals.

05 OBSTACLES TO EDUCATING SALESFORCE DEVELOPERS ON SECURITY

Developers and Admins are the “last mile” of security. Their actions can either reinforce or undermine an organization’s security strategy. It is easy for teams to become so focused on delivering new capabilities that they ignore security considerations.

It is crucial that developers maintain a focus on security to ensure all applications and updates avoid introducing new vulnerabilities to your system. Here are some challenges often seen when trying to get developers to prioritize data security.

5.1. CLEAR-EYED UNDERSTANDING OF SALESFORCE’S SHARED RESPONSIBILITY MODEL

When leaders and teams fail to understand that Salesforce’s Shared Responsibility Model is a two-way street, it can lead to excessive faith in Salesforce’s responsibility for an Org’s security. This creates a false sense of security that leads to an apathetic attitude towards data security. .

One reason leaders and teams may misunderstand the Shared Responsibility Model is because at its core, Salesforce is a business, not just a platform. Companies engage in marketing to grow their own business, and that often involves selling a story about what they have to offer that is free of context or caveats. Salesforce, of course, is no different than any other company. Part of the story they tell about the Salesforce platform involves their commitment to security. Yet even though experts inside and outside of Salesforce readily acknowledge the existence of serious risks from insecure configurations, it is not part of the story told to their customers.

This makes it more important for leaders and teams to understand it is not just a Responsibility Model, it is a ***Shared*** Responsibility Model. The onus of protection falls on the teams implementing Salesforce, so if development teams assume they do not need to keep data security top of mind, we are here to remind them and their leadership that they do.

5.2. LACK OF SECURITY TRAINING

There is a notable lack of specific security training for Salesforce developers. While general security training is abundant, there are few that focus specifically on Salesforce-specific development. This creates a knowledge gap that many developers will not be able to bridge themselves. Support from management is critical for success.

Similarly, most Salesforce training does not adequately convey how easy it is for misconfigurations to lead to vulnerabilities or breaches. Salesforce's learning community, Trailhead, devotes an entire section to the generic training needed for a cybersecurity career.[23] But explanations of how to secure Salesforce itself are conspicuously absent from those trainings.

Salesforce security training is critical, particularly for citizen developers who lack significant technical and security backgrounds. The great promise of low code is making it easier for nonprofessional developers (aka citizen developers) to build solutions to solve the problems they experience in their own work. Such citizen developers typically have security awareness and training far below that of developers working in other technologies.

Organizations are responsible for helping developers gradually learn the tools and gain the skills they need to build safely. This can require slowing down in the short term to go faster in the future.

5.3. DISCONNECT BETWEEN SALESFORCE AND ENTERPRISE IT

Security is rooted in organizational culture and thus reinforced or undermined by every action taken in an organization. The primary stakeholders responsible for security in an organization are typically the CISO, CIO, and Compliance Officers. But these individuals are typically stretched thin, trying to manage a huge portfolio of responsibilities spanning multiple platforms and technologies.

Salesforce was one of the first business applications that could be purchased and used on the Internet without requiring an organization to configure servers and install software. Salesforce's motto in the early years was "No Software." This was a rallying cry for sales and marketing teams who understood this as a means to bypass the arduous process of working through corporate IT to get a new tool set up. "No Software" really meant "No IT."

It is important to understand why that motto resonated so strongly with business buyers. From the point of view of sales teams, dealing with Enterprise Software was "hell".[24] Systems took a long time to get purchased and installed, updates were

infrequent, and customizations were prohibitively hard.

Undoubtedly there were avoidable inefficiencies in the IT departments of that time. But one reason IT initiatives typically take a long time is because they understand the need to establish security and compliance in whatever systems they build. By Salesforce bypassing the IT department, they set a precedent in the minds of business buyers that there was an alternative that did not require them to involve internal IT experts. "Shadow IT" gradually became the new norm.

As SaaS business applications began to proliferate, CIOs came to accept that they could no longer constrain all business applications, and their role shifted to more of a stewardship role, enforcing security reviews during the purchasing process. The assumption was that these business applications bore the responsibility to secure the data contained within them.

As Salesforce grew more complex, its initial configuration began to require teams of experts. Third-party consulting firms stepped in, with Sales and Marketing departments allocating their own money to hiring these companies, outside of the IT budget.

One characteristic of using system integrators to configure Salesforce is that the implementation is treated as a one-time project. As much scope as possible is squeezed into the available budget, leaving little or no room for security analysis and reviews.

Once the project is delivered, it is often maintained by a small team of Salesforce developers and admins. That team may have limited familiarity with what was built by the system integrators. Moreover, because they were not initially involved in either the purchase decision or implementation, IT departments are often oblivious to the details of what has been built in Salesforce. On the contrary, they may be delighted when legacy custom applications can be replaced by applications managed directly by business technology teams.

5.4. DISCONNECT BETWEEN SALESFORCE AND IT SECURITY TEAMS

This gap between IT teams and the business technology teams managing Salesforce also highlights a disconnect between Salesforce and enterprise security teams. Since Salesforce Communities are often not included in the enterprise asset list or integrated with traditional attack surface management tools, they remain outside the scope of these security systems. As a result, Salesforce exists as a large blind spot for security teams, with limited or no visibility into its unique metadata-based architecture.

Most IT security teams do not understand Salesforce and Salesforce security. Often, they focus on traditional security risks that do not apply to Salesforce. For example, initial security reviews focus on the security of the platform itself instead of the customizations project teams have built on top of it. Salesforce passes security reviews disguised as a SaaS application. But the unique vulnerabilities of Salesforce applications are invisible to traditional security tools.

Security teams work in high-stress roles, are typically understaffed, and often battle chronic burnout.[25] After completing their initial security reviews, these teams are often also happy to leave Salesforce in the hands of business technology teams.

Thus, Salesforce development initiatives frequently exist outside of formal cybersecurity architecture review programs within organizations. When formal connections do exist between cybersecurity programs and development teams, the rapid pace of Salesforce customization often overwhelms the security teams' ability to review them, resulting in long delays.

The lack of communication between Salesforce teams and corporate IT security teams presents a significant challenge. This disconnect can result in security vulnerabilities being overlooked until it is too late.

A weak connection between Salesforce teams and corporate AppSec and IT security teams exacerbates the issue. This disconnect leads to IT security teams being undereducated on Salesforce, and Salesforce teams being undereducated on conventional security concerns.

06 RECOMMENDATIONS

Salesforce security is dependent on how the platform is used. Every customization, configuration, and third-party integration introduces an opportunity for security vulnerabilities to pop up. A well-managed Salesforce security program includes education, processes, automated tooling, and a strong partnership between an organization's Salesforce and security teams.

6.1. DISCONNECT BETWEEN SALESFORCE AND IT SECURITY TEAMS

A cultural shift towards prioritizing security is essential for safeguarding systems and preventing breaches. This shift begins with increased awareness and education, fostering a security-first mindset among developers. Security is not just a technical requirement. It is a manifestation of organizational culture: the values, thoughts, and actions of everyone in the organization. Continuous education is vital, as it drives motivation and urgency to implement processes and changes that can have a significant impact.

Over the last 4-5 years, security awareness has improved, with more people sharing knowledge through blogs and community efforts. Salesforce is also continuing to make it easier to build securely on their platform. However, critical obstacles remain, such as insufficient collaboration between Salesforce teams and corporate AppSec/IT Sec teams. It is crucial to promote security awareness and make security practices easier to implement within the Salesforce ecosystem.

The tools, skills, and standards needed to develop securely on Salesforce continue to grow more slowly than the rate of customizations being added to the platform.

Security standards take time to implement. But standards themselves are a form of education, as they give ongoing feedback about what is appropriate and what is inappropriate in different situations. Over time, a team's security understanding will increase, just due to having to comply with security standards and undergo security reviews. Widespread adoption of standards provides a solid foundation for secure operations.

Proactive security planning must consider the human and cultural aspects of implementation, ensuring the effectiveness of awareness and education efforts. While lack of training is often cited as an obstacle, the challenge lies in overcoming resistance and fostering a willingness to embrace security as an integral part of the development process.

6.2. MASTER SALESFORCE SECURITY FUNDAMENTALS

Keep it simple. Any implementation or re-architecture efforts should be guided by the philosophy of keeping systems simple, scalable, and intuitive, ensuring that security measures are both effective and manageable. Implementing secure-by-default configurations is essential to prevent data accidents and ensure that security is embedded in the development process.

Technical debt undermines the value of Salesforce, and security vulnerabilities add far more risk than is apparent. Companies pay Salesforce for a secure platform that will be maintained while remaining flexible and agile. But as complexity grows, organizations lose much of that agility and safety. After several years, companies may even need to re-implement Salesforce, requiring another consulting company or a fresh new Salesforce Org.

While default security settings provide a solid foundation, they often need to be supplemented with more advanced security measures. For example, multifactor authentication and single sign-on can significantly enhance protection for sensitive data.

Implement or use third-party continuous monitoring solutions that track login activities, data access patterns, and export behaviors. Additionally, monitor the use of the “Login As” feature closely and ensure any changes made to configurations or user permissions are thoroughly reviewed before being promoted to the production environment.

Setting Organizational Wide Defaults (OWD) to private by default and carefully layering permissions is a recommended approach to enhance security by ensuring users only have the access they need. This method aligns with the principle of *Least Privilege*, which is often overlooked in small and medium-sized businesses (SMBs), where OWD settings are frequently set to *Modify All* or granted to System Administrator roles.

Emphasizing the *Least Privilege* model serves as the best defense against data exposure. Regular audits should be conducted to ensure permissions are not excessive and that access remains strictly necessary, further strengthening the security posture.

Teams that are unable to follow a DevSecOps approach to team coordination may be limiting their value to the Org. A surprising amount of time and energy can be wasted if the team does not have automated processes to support development, deployment, and security analysis.

6.3. WORK WITH YOUR COMPANY'S SECURITY TEAMS TO DEFINE A PRACTICAL PLAN FOR SALESFORCE

Having defined the risks and potential costs of a security incident and begun to educate yourself and your teams on security, you are now ready to begin creating a security strategy. We strongly advise you to seek help from the security teams already within your organization, and work with them to define a pragmatic strategy for increasing security on Salesforce.

All communication is translation. You will need to translate the unique characteristics of Salesforce to a security team who is mostly oblivious to this platform. And they will need to help you understand the diversity of security concerns that Salesforce might present. Together you will need to do the following:

1. Prioritize risks (what data, processes, or integrated applications require the most protection)

2. Identify potential vulnerabilities in those high-value systems

3. Define an action plan to mitigate any existing gaps in those systems

4. Determine how to maintain the security of those systems over time

The relationship between Salesforce teams and corporate IT is the foundation for protecting internal systems. On that basis, seek out trained Salesforce security specialists to help and provide advice. Such experts are currently rare, but it is our sincere hope that this unique skill set will become increasingly prevalent in the years to come. The risks could not be higher.

Establishing ongoing security protections is far more practical if you have automated tools to help in the detection and remediation of vulnerabilities. Tools are not a substitute for vigilance from the team, but both Salesforce teams and security teams are typically overburdened. It is unreasonable to expect teams to catch all risks unless you equip them with tools that make that detection far, far easier at scale.

CONCLUSION

Securing Salesforce is a complex but critical task. By addressing common vulnerabilities, understanding the major risks, overcoming educational challenges, and implementing expert recommendations, organizations can significantly improve their Salesforce security posture. This paper serves as a guide to understanding and mitigating the key problems in Salesforce development security, helping organizations protect their data and maintain trust.

APPENDIX 1:
LOW-CODE SECURITY ALLIANCE CONTRIBUTORS

The Low-Code Security Alliance (LCSA) began as an informal network of 15 of the leading Salesforce and security specialists in the industry. Motivated by a shared perception that security on low-code platforms such as Salesforce is dangerously underemphasized, the LCSA works to promote education and awareness of this topic.

FOUNDING MEMBERS

The views expressed in this guide represent the collective consensus of the founding members of the Low Code Security Alliance. These views do not necessarily reflect the views of their employers or other associated parties.

Charan Akiri is a seasoned expert in the DevSecOps domain, currently working at Reddit. In 2023, he discovered a critical Salesforce misconfiguration that was leaking sensitive PII data. Charan took the initiative to collaborate with more than 15 government and private organizations to address the issue. He not only developed a comprehensive remediation guide but also provided hands-on assistance to these companies in resolving the problem. In addition to this work, Charan continues to support nonprofit organizations by sharing his expertise in Salesforce security best practices, helping them safeguard their data effectively.

Blanca León-Carter has made significant contributions to the IT industry with a focus on Salesforce and DevOps. She is the VP of Community Engagement at RAD Women, a nonprofit organization providing advanced admins with the foundational building blocks to learn to code Apex. Previously she served as a DevOps Architect on a Global Enterprise Architect Salesforce Delivery team and is a Salesforce Leader with more than 10 years of experience. Blanca led complex implementations in nonprofit, higher education, and public sectors, specializing in enhanced system performance and resiliency across multi-cloud environments. She has evolved her expertise to include Microsoft and Google. As a Well-Architected Ambassador, she leads the Chicago Architect User Group, conducting workshops and training sessions to empower diverse professionals.

Amit Chaudhury is a Salesforce Application and System Architect who has been working on the Salesforce platform since 2010. He has been a Salesforce MVP since 2017 and holds 24+ Salesforce certifications. As an active blogger and speaker at Apex Hours, Amit is dedicated to helping the Salesforce community.

John Crimmings is a Senior Principal and Technical Architect at Slalom. He is an evangelist for source-driven delivery of Salesforce custom code and configurations and has helped companies rethink and rebuild the way they deliver on the Salesforce platform. From this experience, John has become a strong advocate for a more efficient, intuitive, and persona-driven Salesforce security architecture.

Andrew Davis is Chief Product Officer at AutoRABIT. A long-standing Salesforce architect and developer, he is the author of Mastering Salesforce DevOps, the leading book on the Salesforce development life cycle. Building on his passion for the human side of software development, he wrote Flow Engineering (IT Revolution Press), a book on optimizing team workflow and psychological flow.

John Daniel is the Senior Director for Digital Platforms at Steampunk, a prominent Salesforce partner within the federal sector, including agencies such as USDA and CDP. With nearly 15 years of experience in Salesforce, John has held various roles across professional services, product development, and enterprise architecture. His extensive expertise includes serving as the lead architect for U.S. Customs and Border Protection, overseeing all Salesforce development initiatives. John's career highlights a strong focus on Salesforce DevOps and DevSecOps, emphasizing automation and efficiency within development processes. He has contributed significantly to the field through his role as a technical reviewer for numerous books and by managing several open-source projects and frameworks.

Rakesh Gupta has more than 13 years of experience in the Salesforce ecosystem, working on high-security projects for government agencies and various industries for IBM. As a Senior Solution Architect and Manager at IBM, he leads teams to deliver complex solutions, earning recognition as a 10x Salesforce MVP and MVP Hall of Fame member. Rakesh is Level 2 CJIS certified, with expertise in Salesforce Flow, CRM Analytics, and Apex. He cohosts Automation Hour, contributes to Salesforce literature, and coaches on Salesforce courses, aiming to empower professionals to leverage Salesforce for maximum impact and security.

Gaurav Kheterpal, based in Jaipur, India, is the founder and CEO of Vanshiv Technologies, a consulting firm specializing in Salesforce implementations. With 24 years of experience in the IT industry, Gaurav has led numerous security initiatives, resolving critical issues and ensuring data integrity. His extensive background includes managing complex projects where he identified and mitigated security vulnerabilities.

Joshua Kohl is the Senior Manager of Security Enablement for a Salesforce Center of Excellence. In this role, he collaborates across the enterprise to elevate the security and

governance standards for Salesforce products. He has more than 12 years of experience working with the Salesforce ecosystem and more than 25 years of experience leading technical teams across various disciplines, including security and governance, risk, and compliance.

Jason Lord is a cybersecurity executive and subject matter expert with more than 25 years of expertise in managing enterprise security risk, developing programs focused on cybersecurity operations, cloud security, cyber intelligence and insider threat, incident response and handling, and penetration testing. Formerly the Chief Information Security Officer at the White House and a Departmental CISO at Bridgewater Associates, he is currently the Chief Information Security Officer of AutoRABIT.

Matt Meyers is a Salesforce CTA, CEO, and Founder of EzProtect, a virus-scanning solution for Salesforce. With 18 years in the Salesforce ecosystem, he's also the author of the Amazon bestseller Securing Salesforce Digital Experiences. Matt is passionate about well-architected, secure Salesforce implementations and developing the next generation of Salesforce architects.

Matt Pieper is the Director of Engineering, Enterprise Applications, at LeafLink, where he also serves on the security team. Matt is working on building high-performance teams focused on business technology. Matt has worked on the Salesforce platform since 2012 and has deep experience in software development, business intelligence, and multiple other domains.

Jason Ross, founder of Omni Cloud Consulting, has been immersed in the Salesforce ecosystem since 2008. He currently serves as Senior Technical Architect for the Department of State and Senior Salesforce Security Architect for Pentagon Federal Credit Union. Jason's career also includes roles as Chief Information Security Officer, Chief Information Officer, and Chief Operating Officer across various organizations.

Nicolas St-Pierre is the Chief Technology Officer at Agilicus, a cybersecurity company, and serves as an advisor to multiple startups and companies. With 20 years of experience in security for mobile and cloud spaces, Nick has authored patents in 5G security and privacy-by-design initiatives. His expertise spans low-code platforms and the intersection of cybersecurity with Salesforce applications. Nick emphasizes the importance of practical security frameworks and has been instrumental in advancing security practices across various technology infrastructures.

APPENDIX 2:
RESOURCES FOR LEARNING MORE ABOUT SALESFORCE SECURITY

SECURITY-RELATED GUIDES FROM SALESFORCE

Salesforce Security Guide -
https://resources.docs.salesforce.com/248/latest/en-us/sfdc/pdf/salesforce_security_impl_guide.pdf

Salesforce Security Tips for Guest User Access Controls -
https://compliance.salesforce.com/en/documents/a006e00000yTRBFAA4

Salesforce Security: Commitment to Customer Trust -
https://org62.my.salesforce.com/sfc/p/#000000000062/a/3y000000UhUB/wJ940m-JvwK_JEKGhKGcRw8cilzFFrGavpj2L447NtpU

Salesforce Secure Development Lifecycle Overview -
https://compliance.salesforce.com/en/documents/a006e00000yB7cmAAC

Salesforce Enterprise Security Overview -
https://compliance.salesforce.com/en/documents/a006e000010ngGsAAI

Security Perspective on the Shared Responsibility Model:
A partnership in data protection -
https://compliance.salesforce.com/en/documents/a006e000010nW2FAAU

RELATED BOOKS

Davis, A. (2019). *Mastering Salesforce DevOps: A Practical Guide to Building Trust while Delivering Innovation.* Apress.

Jyoti, D., & Hutcherson, J. (2021). *Salesforce Architect's Handbook: A Comprehensive End-to-End Solutions Guide.* Apress.

Malmqvist, L. (2022). *Salesforce Anti-Patterns: Create powerful Salesforce architectures by learning from common mistakes made on the platform.* Packt Publishing Ltd.

Meyers, M. (2024). *Securing Salesforce Digital Experiences*. Palmetto Publishing.

ENDNOTES

1 https://www.gartner.com/en/newsroom/press-releases/2022-12-13-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-20-percent-in-2023

2 https://s23.q4cdn.com/574569502/files/doc_financials/2024/ar/salesforce-fy24-annual-report.pdf

3 https://s23.q4cdn.com/574569502/files/doc_financials/2005/fy05_annual_report.pdf

4 https://www.masonfrank.com/insights/salesforce-careers-and-hiring-guide/

5 https://krebsonsecurity.com/2023/04/many-public-salesforce-sites-are-leaking-private-data/

6 https://help.salesforce.com/s/articleView?id=sf.security_health_check.htm&type=5

7 https://www.salesforce.com/blog/salesforce-ecosystem-explained/

8 https://help.salesforce.com/s/articleView?id=release-notes.rn_experiences_secure_roles.htm&release=248&type=5

9 https://help.salesforce.com/s/articleView?id=release-notes.rn_networks_reduce_object_perms.htm&release=228&type=5

10 https://developer.salesforce.com/docs/atlas.en-us.communities_dev.meta/communities_dev/communities_dev_security_guest_users_apex_access.htm

11 https://developer.salesforce.com/docs/atlas.en-us.communities_dev.meta/communities_dev/communities_dev_security_guest_users_flows.htm

12 https://help.salesforce.com/s/articleView?id=sf.security_sharing_owd_default_settings.htm&type=5

13 https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html

14 https://medium.com/@jonathanbouman/leaked-salesforce-api-access-token-at-ikea-com-132eea3844e0

15 https://www.linkedin.com/pulse/salesforce-certifications-we-getting-them-right-2023-mark-jones/

16 https://genai.owasp.org/llmrisk/llm09-overreliance/

17 https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

18 https://www.ibm.com/reports/data-breach

19 https://www.researchgate.net/publication/356449658_A_meta-analytic_investigation_of_lean_practices_and_their_impact_on_organisational_performance

20 https://www.researchgate.net/publication/211395774_Transformational_Leadership_and_Performance_Across_Criteria_and_Levels_A_Meta-Analytic_Review_of_25_Years_of_Research#:~:text=Based%20on%20117%20independent%20samples,task%20performance%20across%20most%20study

21 https://journals.sagepub.com/doi/abs/10.1177/0021886320920366

22 https://dora.dev/capabilities/transformational-leadership/

23 https://trailhead.salesforce.com/en/career-path/cybersecurity/

24 https://www.salesforceben.com/salesforce-history/

25 https://www.secureworld.io/industry-news/battling-burnout-ciso-cybersecurity