valence

valence threatLabs

# 2023 State of SaaS Security

valencesecurity.com

# Table of Contents

# About This Report

Software has eaten the world, with SaaS at the center of this trend. It was inevitable then, that securing SaaS would become a priority for security teams. Particularly now, since SaaS attacks have been on the rise in the past 2 years. Attackers are engaging in OAuth token abuse, MFA fatigue, and taking advantage of misconfigurations to gain unauthorized access to business-critical SaaS applications such as GitHub, Microsoft 365, Google Workspace, Slack, Okta, and more.

This report explores the past, present and future of SaaS. It is essential to understand how it impacts business, to understand how security should address SaaS-related risks. Exploring the perspective of malicious adversaries is also an important part of the picture we explore in this report. Most interesting, perhaps, are the insights we've gleaned from working with our own customers - understanding their challenges and how Valence can address them in the most labor-efficient way possible.

Finally, we wrap up the report with fourteen recommendations and three predictions. We hope you enjoy this report and find some valuable and actionable insights in its pages.

# Why SaaS Security?

It's true that security teams have to worry about vulnerabilities, misconfigurations, and other security issues across a wide variety of environments: cloud, on-prem data centers, remote employees, contractors, and third parties, to name a few. How should SaaS be prioritized?

SaaS has come a long way since the birth of Web 2.0. In the early 2000s, companies like Salesforce and Qualys were pushing the boundaries of what could be done in a web browser. These days, SaaS is ubiquitous, and it's unremarkable for product onboarding to take less than an hour. In 2023, it's difficult to justify still running email or file-sharing services on-prem. The headache of managing and protecting most software functions on-prem just doesn't make sense anymore.

Over the past two decades, SaaS applications have also evolved into complex platforms. SaaS vendors like Google and Microsoft now allow users to share, automate and collaborate within their platforms, with the goal of minimizing the need to ever leave their platforms. The SaaS ecosystem has evolved into a complex mesh of dozens of interconnected applications.

Another relevant trend is the decentralization of SaaS adoption and use. Twenty years ago, if employees needed a new software platform, they'd need to submit a request. IT would do POCs, buy hardware for the new platform, deploy the hardware and install/configure the software. Months later, employees might have access to their new software platform. Today, this process takes minutes and doesn't require much, if any, technical expertise. The result is that employees outside IT are deploying and managing complex software platforms, often without the IT or security group's knowledge.

As businesses migrate their most critical business functions to SaaS platforms, they're often not aware of how attack surfaces or security options have changed. SaaS platforms represent new security challenges. Attackers take advantage of insecure defaults. The work necessary to harden these environments against damage from attacks or neglect is not obvious.

# More SaaS Adoption → More SaaS Breaches

Attackers are always developing new attacks and strategies. Over the past few years, they've increasingly turned to stolen credentials and SaaS as an entry point. The move to SaaS has created a dichotomy: as multi factor authentication use increases (high friction, difficult to bypass), so does the use of authentication tokens for human and non-human identities (low friction, trivial to steal). A stolen token can log an attacker in without needing to know the username, password, or any second-factor authentication - the token bypasses all factors.

**The following sections will explore the methods used by attackers to breach SaaS applications, along with some prominent examples for each method.**

## Threat 1 Raiding the Cookie Jar

As previously mentioned, tokens stored in browser cookies are highly valuable to attackers. They will often log the attacker in with no prompting for passwords or secondary authentication factors. Specialized malware (info stealers) exists to gather up these credentials, and specialized markets exist to buy and sell them.

Stealing sessions is often as easy as `chrome.cookies.getAll({})`

The Slack credentials that led to the June 2021 Electronic Arts breach cost the attackers $10 on the Genesis Market.

These tokens exist for one simple reason: to make SaaS easier to use. Asking users to log in every time they want to use a service is considered to be an unacceptable level of friction for most applications today. Imagine having to log into nine different Slack instances once an hour, all day long.

Initially, only consumer SaaS services issued tokens that would keep users logged in for months or years. The consumerization of IT trend brought consumer design principles to enterprise SaaS, and this 'log in once' principle came with it.

### Example: CircleCI SSO Session Cookie Theft

We've now seen a few attack campaigns that begin with targeting an engineer with malware to steal corporate credentials. In the case of CircleCI, information-stealing malware infected an engineer's laptop. Among the credentials scooped up by this malware was a valid, 2FA-backed SSO session that gave attackers access to CircleCI's private GitHub repos.

Logged on as the engineer, the attacker was able to generate access tokens to production CircleCI environments, which then led to the abuse of Github machine-to-machine OAuth tokens belonging to CircleCI customers.

In a very similar attack, Slack employee tokens were stolen and used to access the company's private GitHub repositories as well. Luckily, in the Slack case, access to the source code didn't lead to the theft of customer data. One of the most concerning features of these attacks was that MFA didn't hinder the attacker - the engineer's stolen session token worked without reauthentication. A positive outcome is that these incidents have led to more awareness that not all multi-factor authentication is equal in strength and resilience.

# Threat 2 Battlebots - SaaS-to-SaaS Token Abuse

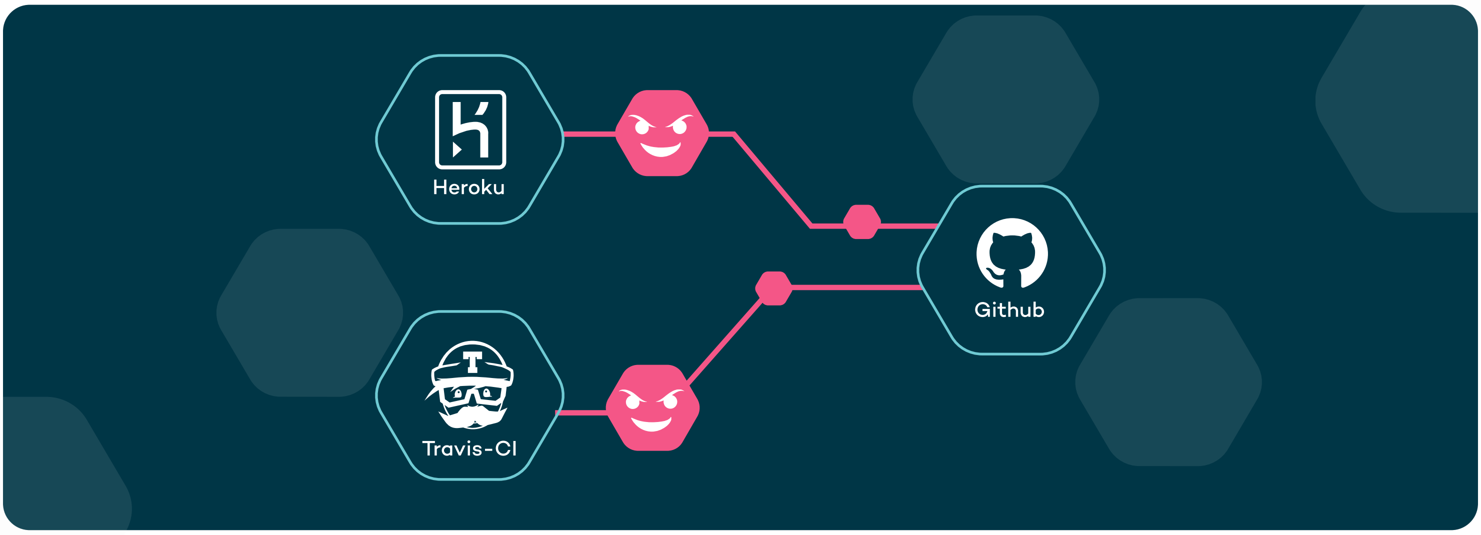
Tokens aren't just created when users log into a SaaS application. Sometimes an automated service needs access to SaaS as well. These tokens are both very powerful and very easily abused. Once again, no username, password, or multifactor authentication is necessary to use them. Once stolen, they just work. Since they're typically used to access the service via an API, there's typically no indication to the user or the vendor that the token was stolen and is being actively abused.

### Example: Heroku and Travis-CI GitHub Tokens Abuse

In this incident, attackers targeted machine-to-machine (SaaS-to-SaaS) OAuth tokens. They gained access to Heroku and Travis-CI's secrets, which contained GitHub OAuth tokens that were granted to them for legitimate purposes. The attackers then abused the tokens and gained unauthorized access to dozens of GitHub organizations and downloaded data. This included GitHub's own private npm repositories, which also led to the discovery and abuse of AWS API keys and access to AWS S3 storage.

This tangled web of dependencies and interrelated credentials is not uncommon. Even with the benefit of hindsight, making sense of how the incident occurred is difficult, due to the complicated nature of integrations between GitHub, Heroku, and Travis-CI.

> It took all three organizations weeks to reset all the credentials connected to this incident before they were finally able to declare the breach over. It took another 20 days before Heroku was comfortable enough to re-establish GitHub integrations for their customers.



## Threat 3 Out of Sight, Out of Mind

Attackers know many SaaS applications are left unmonitored and happily take advantage of this oversight. The same old noisy attack methods that worked in the past on internal corporate networks continue to work in the SaaS space for the same reasons. Creating privileged accounts and/or adding accounts to privileged groups should never go unnoticed - SaaS or otherwise.

### Example: Okta Third-party Vendor Compromise

This attack, by the LAPSUS$ hacking group, didn’t compromise Okta directly, but compromised Sitel, a third-party Okta used to provide support to customers. Regardless, as a company hired by Okta to work with its customers, the ultimate responsibility lies with Okta. The attack was multi-faceted, but there were many opportunities to detect and stop the attack from a SaaS Security perspective.

- Attackers created email forwarding rules to send copies of mail to accounts controlled by them
- They gave themselves additional privileges (added to TenantAdmins group)
- They accessed and created Microsoft 365 accounts
- Attackers found plaintext credentials in Sharepoint (DomAdmins-LastPass.xlsx)

| | | |
|---|---|---|
| 2022-01-21 00:05:15 | [ACCOUNT NAME REDACTED]@sykes[.]com accessed hxxps://[INTERNAL URL REDACTED]/personal/[INTERNAL USER NAME REDACTED]/Documents/Proiects/ryk/DomAdmins-LastPass.xlsx via SecureLink | **Internal Recon** |
| 2022-01-21 05:29:50 | [ACCOUNT NAME REDACTED] account created by [ACCOUNT NAME REDACTED]@sykes[.]com | **Maintain Presence** |
| 2022-01-21 05:29:51 | [ACCOUNT NAME REDACTED] added to TenantAdmins group by [ACCOUNT NAME REDACTED]@sykes[.]com | **Maintain Presence** |
| 2022-01-21 05:39:13 | Malicious Email Transport rule to forward to BCC all mail to the accounts [ACCOUNT NAME REDACTED]@sykes[.]com and [ACCOUNT NAME REDACTED] | **Establish Foothold** |

*Partial Timeline of Sitel Investigation Showing Undetected Microsoft 365 Abuse

Threat 4

# Wolves in App Clothing

In a particularly damaging type of campaign, sometimes attackers use a legitimate company's trust and reputation to trick and attack others. It makes sense from an attacker's perspective. Taking advantage of an existing company's hard-earned trust and reputation is much easier than building a fake company from scratch.

In many of these attacks, the bad guys take the time to understand context and relationships. This makes attacks more convincing and allows for precise timing. We've also seen attackers create fake applications in an attempt to lure users into giving the attackers broad access to the customer environment.

The impact can't be overstated: if attacks could come from any legitimate identity or organization, spotting an attack becomes much more difficult. The reputational damage is difficult to recover from as well.

### Example: Malicious Apps and the OiVaVoii Campaign

The OiVaVoii campaign isn't a single breach, but an attack campaign that used a novel method to use the reputation of one organization to attack others. Attackers compromised Microsoft 365 tenants and used this access to create malicious OAuth apps. These apps were then used to target other organizations using a mix of social engineering and legitimate apps leveraging a compromised corporate identity. This created a whack-a-mole situation where Microsoft had to try to block the malicious apps as quickly as possible as they popped up, not knowing which legitimate customers were compromised.

Threat 5

# Marvelous Misconfiguration Misadventures

Leaky AWS S3 buckets are old news, but data leaks can occur due to SaaS misconfigurations as well. Misconfigurations happen for a variety of reasons:

1. Lack of familiarity with configuration or sharing options and their effect
2. Insecure defaults
3. Poor UI/UX (I thought this setting meant 'public to my organization', not 'public to the Internet')
4. Poor, or missing documentation
5. Urgency leads to sloppy work; skipped security checks and testing
6. SaaS platforms lacking the ability to monitor for unauthorized access
7. Employees without IT or security training managing configuration/sharing settings

Regardless of the reason, misconfigurations are responsible for a large number of unintentional data breaches. Attackers know how common these gaffes are and how to find them, often automating the job of searching for them.

### Example: Leaky Salesforce Sites

Researcher Aaron Costello discovered how easy it was to misconfigure Salesforce Community sites. It was possible for unauthenticated website visitors to access sensitive data that should have only been accessible to authorized users. In 2020, he published a detailed write-up on how to exploit, fix, and monitor this misconfiguration. That wasn't enough to prevent organizations from continuing to misconfigure their Salesforce sites, however.

Security Researcher Charan Akiri discovered numerous government and private organizations with misconfigured Community sites and attempted to notify them. He later contacted Brian Krebs to help get the word out about this issue, as government agencies ignored his warnings.

> A number of problems led to this issue becoming widespread. Many organizations built websites hastily during the pandemic, for assistance programs and other pandemic-related needs. The misconfiguration was not obvious to those building these sites. Salesforce did not make it easy to monitor for abuse of this misconfiguration.

# Valence SaaS Security Findings

We peered across customer environments to get a snapshot of what 'normal' looks like in organizations before our SaaS security platform was leveraged to clean things up. The data in this section represents 12 different industry verticals and hundreds of thousands of users. The snapshots are focused on the productivity suites only. For the sake of simplicity and comparability, each snapshot used in this analysis is taken from either Google Workspace or Microsoft 365. The data has been aggregated and anonymized to ensure customer privacy.

## The Great SaaS Garbage Patch

### Ungoverned SaaS Use Generates Massive Mess

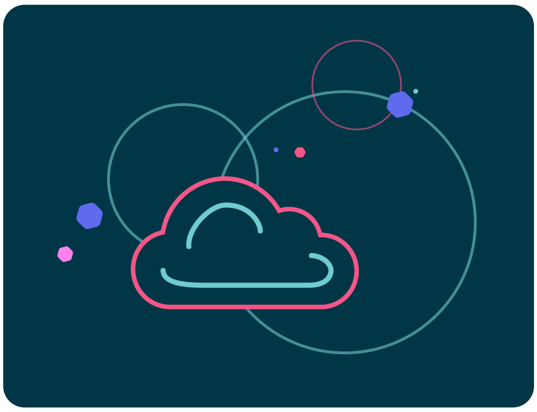
Early adopters will often try out several different SaaS apps.  When they settle on one, the ones that didn't make the cut are often forgotten about and are left connected. The result is a pile of junk that builds and builds over time!

On average, over half (51%) of an organization's SaaS third-party integrations are inactive.

90% of an average organization's shared assets (e.g. files or folders shared with external collaborators or to "anyone with the link") hadn't been accessed for at least 90 days.

**IMPACT:** more SaaS equals more risk - when there's a benefit to using risky apps (like an increase in productivity), that could be a risk worth accepting, but with abandoned SaaS integrations and idle data sharing, however, there's no benefit.

**RECOMMENDATION:** regularly remove unused SaaS integrations and revoke idle sharing to reduce risk and attack surface.



## Everything, Everywhere, All at Once

### Tenant-Wide SaaS-to-SaaS Integrations Are Regularly Abandoned

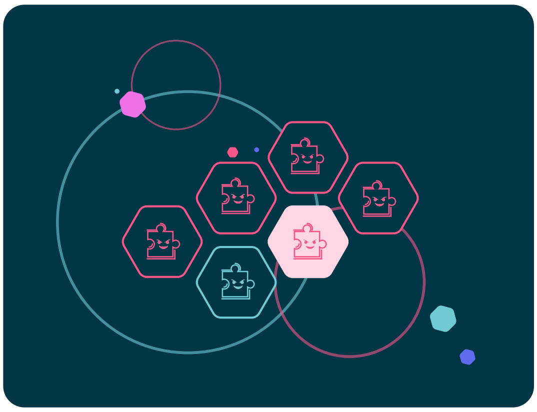
Sometimes, SaaS administrators give third-party integrations access to everything. This gives third parties full read-write access to email, files, source code and more. The multitude of GitHub breaches last year is an example of what can go wrong with broadly granting tenant-wide access. Most security teams don't have the ability to continuously review and right-size overly-broad permissions for their integrations that were set up incorrectly or drift over time.

100% of organizations have granted full read/write access to email, files, and calendars to at least one 3rd party.

On average, there were 21 integrations per organization with tenant-wide access to company and employee data.

**IMPACT:** a single compromised token at this level could compromise all employee email, files, and calendars or leave the organization open to SaaS supply chain attacks.

**RECOMMENDATION:** sunset unused tenant-wide integrations immediately. Ensure the remaining ones are vetted well-protected and closely monitored.



## Great Power, Little Responsibility?

### Without Visibility and Guidance, Data Sharing Quickly Gets Out of Control

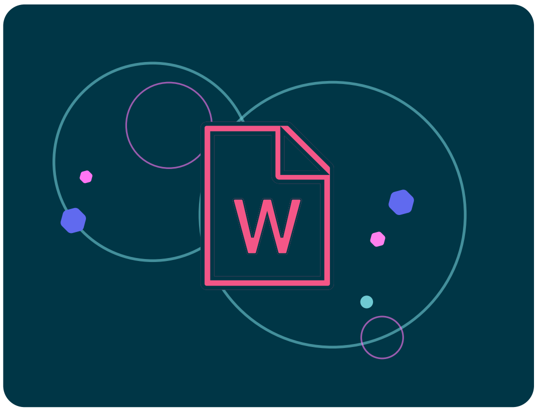
Employees often don't realize or appreciate the power they have. They are simply getting the job done but the risks of oversharing data externally often aren't clear to them. In many cases, it takes only a few clicks to put massive amounts of data at risk through file and data sharing (via GDrive, OneDrive, Box, etc).

30% of the time, files are shared with personal accounts.

There are 54 shared resources (e.g. files, folders, Sharepoint sites) per employee, and 193,000 shared resources per company, on average - most sitting idle and unused.

**IMPACT:** the business loses control of its data and access to data is no longer limited to just employees.

**RECOMMENDATION:** overuse of asset sharing is often due to poor processes or workflow; train employees on alternatives to public links; many data shares are only used once, so consider automatically revoking them after a short period of time (e.g. 30 days).



## Sleepy Accounts, Active Risks

### Even After Employees Leave, Their Actions Create Risk

Dormant employee accounts can be a source of serious risk, especially when misconfigured. They're easy to overlook, and risky employee actions can remain, even after accounts are removed.

On average, 1 in 8 employee accounts are dormant (and as high as 1 in 3 in some companies).

On average, 10% of an organization's shared integrations and data can be traced back to ex-employees.

**IMPACT:** attacks often target dormant accounts, as there is less focus and visibility on them, and there's a lower chance that they're compliant with current security policies.

**RECOMMENDATION:** Lifecycle management applies here as well - organizations should update identity sunset processes to consider an employee's SaaS footprint, beyond just the identity, as they exit the company. Will deactivating their account break critical business processes? Is it possible for ex-employees to continue to access company resources, even after their accounts are deactivated?

## Death by Exception

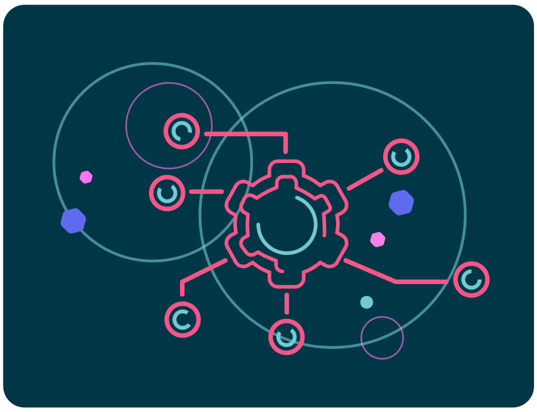
### 99% Secure Doesn't Count For Much If It Doesn't Make The Remaining 1% More Difficult to Attack

While issues with integrations and file sharing could be seen as 'death by a thousand cuts', it often only takes one misconfiguration to let an attacker in. Accounts that require exceptions (e.g. service accounts and guests) make it easy to overlook active user accounts missing basic security, like MFA enforcement.

In most tenants, MFA wasn't enforced by default for all user accounts leading to at minimum 1% of accounts without proper MFA configuration

**IMPACT:** Even as low as 1%, we've seen over and over, credential stuffing via a single misconfigured account can lead to total compromise

**RECOMMENDATION:** Whether approved as an exception, or overlooked as a misconfiguration, identities are one of the most common entry points for attack, and therefore justify closer scrutiny. Ensure MFA is both enabled and enforced for humans, and that privileges and access are minimized for machines.



# Top 14 SaaS Security Recommendations

## SaaS-to-SaaS Integrations

1 Review new integrations as employees onboard them. Consider vendor reputation, configuration options, and monitoring options.

2 Offboard any unused or unnecessary integrations on a regular basis and ensure zero trust principles are enforced to apply least privilege.

3 Continuously communicate with business users to understand the business context and current use of third-party integrations to ensure validation of the need for new, existing and inactive integrations.

## Identities & Permissions

4 Closely manage accounts with high privilege and admin access and apply least privilege principles to ensure each user has the minimum required permissions.

5 Ensure SaaS account deactivation is included in identity lifecycle processes; investigate idle accounts and deactivate if the employee has left the organization.

6 Use SAML, SSO, IdP or at a minimum, enforce strong and unique passwords for all SaaS accounts and enable multi-factor authentication.

## External Data Shares

7 Implement data labeling based on the company's data classification policy. This will help employees understand the rules regarding external data shares.

8 Regularly monitor and review external data shares, or consider blocking shares for employees with clearly defined needs.

9 At a minimum, monitor and review email forwarding, particularly when it is directed to private email accounts.

## Misconfigurations and Compliance

10 Investigate how to leverage native security controls embedded into each SaaS application and configure them according to industry best practices based on standards from NIST, CIS, and CSA.

11 Monitor for configuration drift and collaborate with business application owners to understand the reasons for drift.

12 Determine which compliance requirements apply to SaaS applications; monitor and adjust as needed

## Threat Detection

13 Ensure maximum coverage and analysis of SaaS applications events, activities and admin logs, to detect anomalous and malicious activities.

14 Study SaaS breaches to understand how they occur and incorporate lessons learned into security event monitoring processes.

# Three SaaS Security Predictions

Generative AI took center stage as a motivating force for both SaaS growth and startup funding in 2023. Valence Threat Labs predicts a particular need for diligence over the next year, as excitement about productivity gains and fear of getting left behind by competitors causes businesses to rush in.

Valence Threat Labs recommends that security teams pay close attention to this fast moving trend to ensure they're prepared to partner with the business to understand use cases and quickly adapt to leverage this new technology in the safest way possible. These are our top predictions for the coming year:

### SaaS security will need to evolve beyond visibility to include automated remediation.

Traditional SaaS security solutions have focused primarily on providing visibility into security SaaS adoption and use, with some light capabilities around risk mitigation. Overworked and under-resourced security teams need tools that not just uncover risks but help remediate them. The impact of automated remediation can't be understated - it's the difference between adding more noise and removing it.

SaaS is an ideal environment for automated remediation as well. Use cases are clear, with minimal downside. The worst case scenario for a policy that automatically unshares files, for example, is that the user simply reshares it. As automated remediation is proven to work in SaaS, it will become a must, rather than a nice-to-have for security teams in coming years.

### SaaS security will need to be addressed as a collaborative effort.

One of the most compelling trends in security is often referred to as 'user-focused security' or 'collaborative security'. It sounds obvious once it is said, but security teams can't secure their goals without help from other employees. Just-in-time education on secure practices helps employees understand the impact their choices have.

As users adopt SaaS applications independently, the value of relationships and collaboration between users and security teams increases. An additional benefit to this collaboration is the opportunity to better understand business processes as well. The better security teams understand business context, the more valuable they will be to the business. This collaborative, user-focused approach allows the business to continue moving at a competitive pace without leaving security behind.

### Generative AI will produce an enterprise SaaS adoption boom

Enterprises might be hesitant in this early stage of generative AI use, but a new wave of SaaS powered by AI is on the horizon. Funding is flowing and founders are building. Adoption will likely be decentralized, like most other SaaS use we see today.

It will be more important than ever to understand the SaaS mesh as every organization begins to make decisions about how these emerging tools can and should be used. Valence Threat Labs sees data security and privacy as one of the chief concerns when adopting new business AI tools - particularly large language models (LLMs). Data classification policies must make clear which data can and cannot be used for training, fine-tuning, and in prompts.

Widespread integrations and low/no-code providers will make security analysis and threat mapping challenging. Just like it is currently common for businesses to accidentally expose private data due to cloud and SaaS misconfigurations, we expect to see private enterprise LLMs accidentally exposed to the general public as well.

valence

## About Valence Security

Valence Security is the first security company to offer collaborative remediation workflows that engage with business users to contextualize and reduce SaaS data sharing, supply chain, identity, and misconfiguration risks with scalable policy enforcement and automated workflows.

With Valence, security teams can secure their critical SaaS applications like Microsoft 365, Google Workspace, Salesforce, and Slack and ensure continuous compliance with internal policies, industry standards and regulations, without impeding business productivity or the speed of SaaS adoption.