Harmony
SaaS

THE CISO'S DEFINITIVE GUIDE TO SAAS SECURITY

CHECK POINT™

TABLE OF CONTENTS

03 Executive Summary: SaaS security can be fixed

04 Managing the Unknown: SaaS Puts your Users and Data at Risk

05 How Prevalent are SaaS Breaches?

06 Common SaaS Breach and Attack

11 Can SSPMs and CASBs Prevent SaaS Breaches?

13 SaaS Security Must-Haves - The Ultimate Checklist

19 How Check Point Harmony SaaS Helps

EXECUTIVE SUMMARY

SAAS SECURITY CAN BE FIXED

SaaS usage within organizations is exploding, with the average number of applications within SMBs, mid-market and enterprises standing at 253, 335 and 473 respectively[1]. SaaS applications are also forecasted to contribute 36% of worldwide public cloud end-user spending according to Gartner.[2]

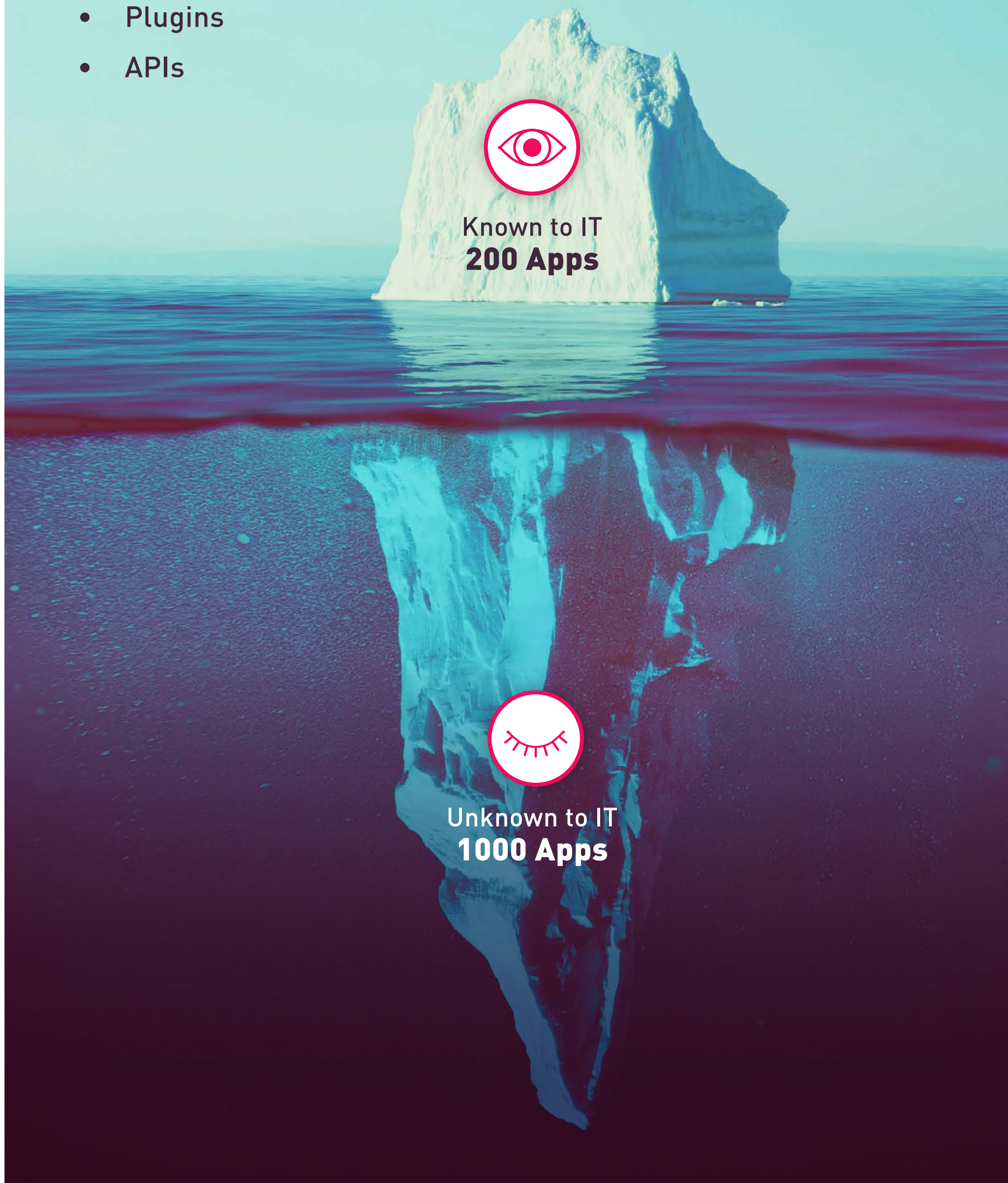
And while these numbers are impressive, it's the SaaS platforms you **don't know about** that tell the whole story. **According to research conducted by Check Point, IT teams are on average only aware of 20% of the SaaS applications being used.** This presents a major challenge for CISOs as they do not have full visibility of their SaaS ecosystem, potentially exposing the entire organization to a major breach as a result of malicious account takeovers or the exposure of sensitive data.

This guide addresses the challenges CISOs face when it comes to securing their SaaS ecosystem, from common SaaS-related breaches to how you can apply proactive prevention against SaaS-based threats.

You know about your SaaS platforms.

But your SaaS platforms are just a fraction of thebroader story.

- Shadow SaaS
- Plugins
- APIs



[1] Source: 60+ eye-opening SaaS statistics (updated for 2024) www.spendesk.com/blog/saas-statistics/

[2] Source: Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach $679 Billion in 2024. November 13, 2023. www.gartner.com/en/newsroom/press-releases/11-13-2023-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-679-billion-in-20240

MANAGING THE UNKNOWN

SAAS PUTS YOUR USERS AND DATA AT RISK

An organization uses dozens of known applications but it's the other SaaS applications you don't know about that put your users and data at risk, including shadow SaaS, plugins and APIs that have not been vetted by IT and security teams and are only protected with minimal default security controls at best.

Complex to Secure

Whether sanctioned or unsanctioned, eemployees' rely on dozens of SaaS applications to collaborate, increase productivity, and perform their daily activities. These apps combine to form a vast ecosystem of shadow SaaS and connected.

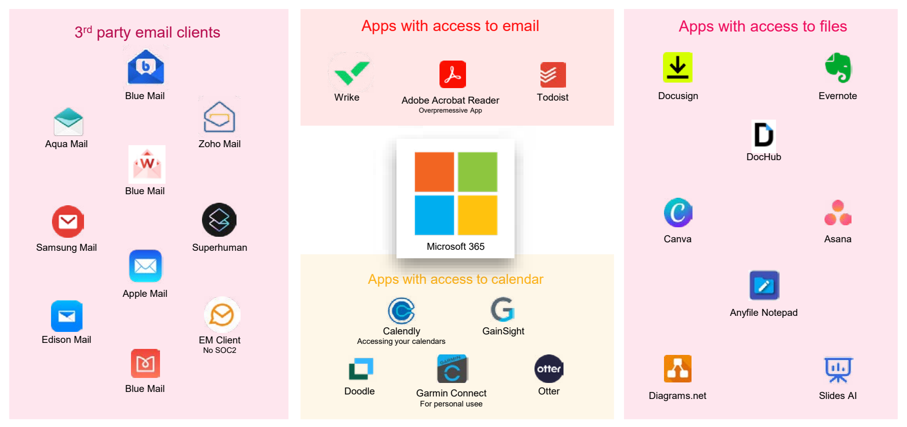
Contains Sensitive Data

Not only do enterprise SaaS platforms contain sensitive data such as personal records, payment data, and source code, but third-party applications often require permissions such as full access to employee calendars, files, and emails, greatly exacerbating the risk of a breach.

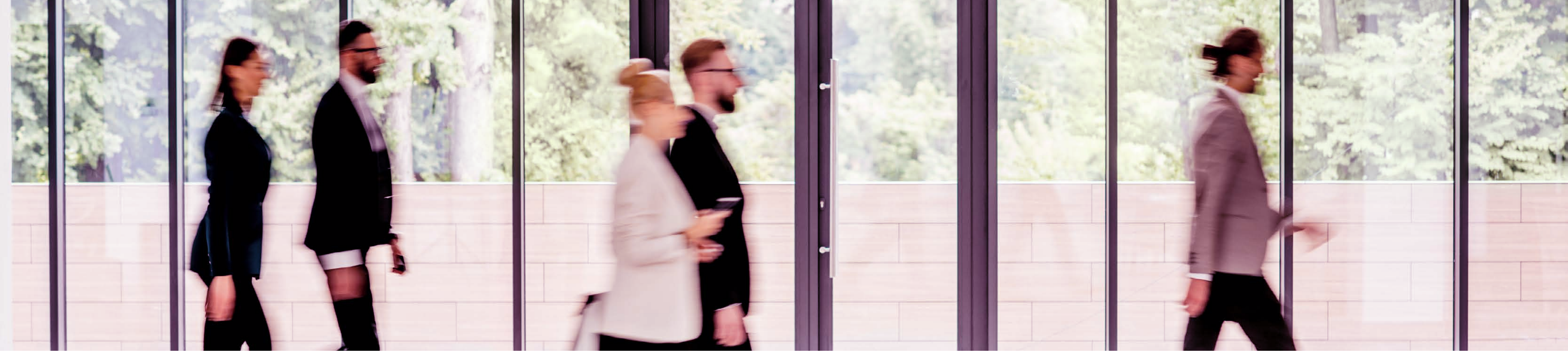
Multiple Threat Vectors

There are numerous ways applications can be compromised, from exposed API keys to excessive permissions given to OAuth-integrated apps, to insufficient user authentication. But essentially, SaaS related breaches boil down to two types to attacks: incidents involving data theft and incidents involving unauthorized access with varying levels of severity, for example access to an account with administrator privileges vs. a standard user account. Industrial cyberespionage can also result from the latter.

Dozens of Applications Have Full Access to Your Data



All the above attributes present a significant challenge for CISOs responsible for overseeing an organization's security posture. These challenges are further compounded by hybrid work, as employee activity takes place outside the confines of the enterprise network firewall and may be only partially monitored, or not at all.



HOW PREVALENT ARE SAAS BREACHES?

SaaS Supply Chain Attacks

With 98% of organizations globally linked to compromised third-party vendors[3], it comes as little surprise that trends show a surge in SaaS-related breaches. CISOs face the seemingly impossible challenge of securing what lies beyond their visibility. According to one study, 50% of organizations have indirect links to 200+ fourth-party breached vendors.[4]

SaaS Data Leakage

Eighty one percent (81%) of organizations experienced data exposure due to SaaS applications[5]. As most enterprise data is held in SaaS, data loss whether intentional or unintentional is a key concern for security and IT leaders, along with the implications on reputation, consumer trust and regulatory compliance fines.

SaaS Misconfigurations

Forty-three percent (43%) of organizations experienced security incidents directly traced to SaaS misconfigurations[6]. Moreover, research shows[7] that in the average company, 157,000 sensitive records are exposed to everyone on the internet through SaaS sharing features, representing $28M in data breach risk. Compound those risks with the time and resources it takes to find and remediate threats post-breach and the costs become even more substantial.

So how do these breaches happen, and what can you do to prevent them in the first place? We'll start with common SaaS breach and attack vectors.

[3] www.cybersecuritydive.com/news/connected-breached-third-party/641857/
[4] www.helpnetsecurity.com/2023/02/02/relationships-breached-fourth-party-vendors/
[5] financesonline.com/top-saas-security-risks-and-how-to-avoid-them/
[6] www.resmo.com/blog/saas-security-statistics
[7] info.varonis.com/hubfs/Files/docs/research_reports/Varonis-The-Great-SaaS-Data-Exposure.pdf

COMMON SAAS BREACH AND ATTACK VECTORS

There are five main types of breach and attack vectors, as described below.

ABANDONED, DEPRECATED AND LEGACY APPLICATIONS

While this SaaS breach vector may seem obvious, identifying applications that should no longer be used and revoking their credentials is not something most companies do on a proactive and regular basis, let alone in an automated fashion.

SaaS services that are no longer being maintained are likely to contain vulnerabilities, while still being connected to your latest source code or customer data. This puts sensitive information at risk of falling into the wrong hands, as the abandoned, deprecated or legacy app maintains access to your enterprise application and its data.

Abandoned Applications

To illustrate this attack vector, let's take an abandoned application as an example. An application that is not maintained is ripe for exploitation like an outdated operating system that is no longer patched. As many SaaS integrations are based on Webhooks, which are explained below, this means that outdated applications may be sending your data to...

someone you don't know about.

How does a Webhook work?

A Webhook means that when something happens in Service A, it sends the data to Service B by sending it to a web address, or a Webhook.
For example, one might run a service called MyCoolService, and when something happens in your GitHub account, GitHub will automatically send the data to hxxps://webhooks.mycoolservice.com/.
Many SaaS integrations in GitHub are based on webhooks and get triggered whenever source code changes in a certain account, resulting in the sending of sensitive information to those webhooks.

Webhooks and abandoned SaaS apps

In one real-life example, a webhook was discovered that was defined for a company that no longer existed (e.g., mycoolservice.com), sending GitHub updates to that ghost service. Moreover, since the company didn't renew its DNS registration (for mycoolservice.com), anyone could essentially have purchased their domain name and claimed it as their own, and then receive the sensitive updates to their newly-claimed website or email address (person@mycoolservice.com) and eavesdrop or sell that data to malicious actors.

STALE API TOKENS AND STALE USERS

Many recent high-profile breaches were carried out by abusing a stale API token or stale user account whose existence was forgotten. Companies often connect a service to their enterprise ecosystem, use it for a while, and then stop. Now you have a connection to your enterprise that has long been forgotten and neglected, and a potential entry point for a threat actor. And when things start looking suspicious it may already be too late.

Additionally, if an abandoned user is hijacked, it will take longer to notice than an active user account, as an active user would notice that something is off sooner.

Such is the case with stale accounts with active API tokens. When an individual leaves a company, admins usually don't investigate what APIs they connected to the organization's SaaS ecosystem. This means that a webhook may still be transferring sensitive data from an internal application to an external one, for example, sending mobile text messages each time a message is posted on a collaboration channel. This gives rise to a situation where the account is stale, the user can no longer log in, but the API token is still active, creating a hazard that a threat actor may find and exploit.

SUPPLY CHAIN ATTACKS

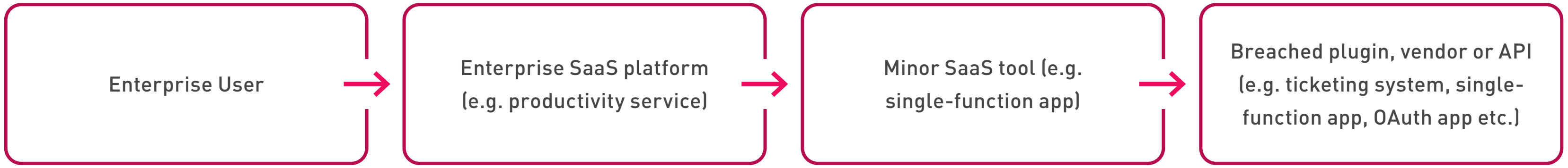
In a third-party or supply chain attack, a SaaS vendor your enterprise is connected to is breached. As a result, the attacker not only has access to the initially-breached vendor's system, but also to anything else the system is connected to.

Lateral Movement through Third-party Attacks

While identity and access management providers come to mind, simpler tools may be abused as well. For example, SaaS tools that help you schedule meetings and have access to your calendar, or a grammar-improvement tool that has access to your email. These services are likely not thought of as security risks as they don't hold much sensitive information, but now that they're breached, attackers obtain access to all the calendars, emails or other resources of the system they were connected to. Similarly, an AI tools that helps design presentation slides may be breached, providing a threat actor with access to all the

Fourth-party supply chain attacks

The above is an example of a third-party supply chain attack, consisting of an enterprise resource connected to an additional breached SaaS service. In fourth party supply chain attacks, a service provider, plugin, or API connected to a SaaS tool is itself breached, so the chain looks something like this:



Enterprise User → Enterprise SaaS platform (e.g. productivity service) → Minor SaaS tool (e.g. single-function app) → Breached plugin, vendor or API (e.g. ticketing system, single-function app, OAuth app etc.)

Example of SaaS Interconnections in a Fourth-Party Supply Chain Attack

MALICIOUS APPLICATIONS WITH A BACKDOOR

Applications may behave maliciously by design, and not as a result of getting hacked. Consider the following scenario:

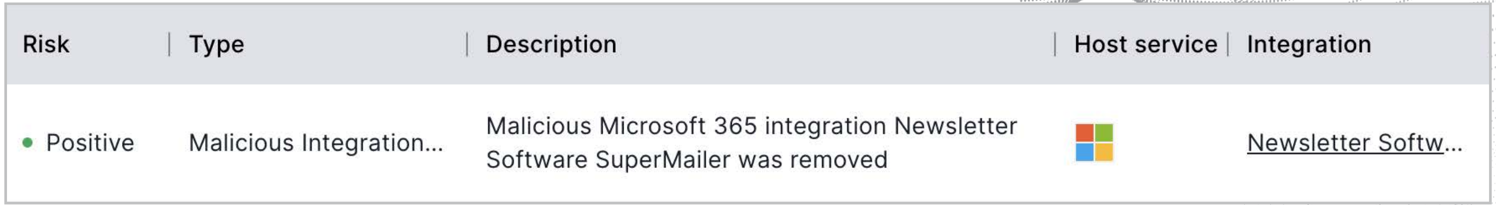
Newsletter App Exploited for Spear-Phishing

A midsize company had a very bad case of phishing emails originating from within the company. Two specific employees were sending well crafted phishing emails to other employees. The security team originally thought that the users were compromised and proceeded to replace their laptops and change their passwords, but phishing continued.

Only after mapping their SaaS-to-SaaS connections did they detect that they have a malicious app installed, a newsletter mailing tool in this case, that was sending emails on their behalf. Using SaaS-specific threat intelligence, the app was identified as malicious elsewhere and was immediately disconnected from the enterprise's mail exchange server. As is generally the case with internal spear-phishing, because these emails were sent from one company employee to another, they were not inspected by an email security gateway, hindering their identification as malicious.

Apps akin to a Wolf in Sheep's Clothing

The app was seemingly legitimate and performed the activity it was supposed to, but it also had a backdoor that could be abused for nefarious purposes, including sending phishing emails, and in this case, it was abused to carry those out. This is just one example of countless others.



Risk	Type	Description	Host service	Integration
● Positive	Malicious Integration...	Malicious Microsoft 365 integration Newsletter Software SuperMailer was removed	🟥🟩🟦🟨	Newsletter Softw...

Malicious app identified and disconnected from enterprise platform

MOBILE CONSUMER APPS WITH EXCESSIVE PERMISSIONS

Mobile consumer apps are a source of significant concern as they often require excessive permissions as a requisite for their use.

A Package Tracking App with Enterprise Access

For example, one high-profile online shopping service requires buyers to download a package tracking app on their phones to track their order's progress (to see when it's shipped, arrived etc.), instead of simply providing a link to their emails.

The result? When installing the app on their phone, the app requires a long list of permissions-with data collected including messages in emails and other in-app messages. If a user is not careful and just clicks "Accept," they have now opened all their enterprise emails and text messages to a shopping service, which is clearly not designed for enterprise use, and has no SOC2 or other certifications.

A Caller ID App that reads your Directory

In another example, an employee connected a caller ID mobile app to their enterprise contacts directory. The app identifies the caller by reading all your contacts and storing them in the cloud, and then showing them to other app users.

The employee installed the app and accepted permissions without noticing that the access it requested was to enterprise contacts, not their personal ones. In this way, the user allowed the app to read the organizations' entire directory including email addresses, phone numbers, job titles and org chart.

Unfortunately, the caller ID app in questions sells people's data, meaning that the company's entire directory was sold, and that attackers can potentially buy this information to execute spear-phishing campaigns against the company.

In summary, while many apps are 'free,' they may cost companies big through data leaks caused by excessive permissions unwittingly provided to them by employees.

ARE SSPMS AND CASBS SUFFICIENT FOR PREVENTING SAAS BREACHES?

SaaS Security Posture Management (SSPM) solutions and Cloud Access Security Brokers (CASB) solutions have emerged to tackle SaaS security from different angles. But can they prevent SaaS-based threats in real time? Let's take a closer look.

CASBs secure User-to-App and In-App Activity

Cloud Access Security Brokers (CASB) help enforce security policies across the enterprise by implementing access controls, uncovering SaaS usage visibility, preventing threats and protecting data. CASBs operate in two complementary modes: API-based security and inline security.

API-based security is installed on sanctioned enterprise SaaS applications that are used on a daily basis, such as G-Suite, Microsoft 365 and Slack, and inline or proxy-based security is used to secure the long tail of shadow SaaS and IT applications that are not sanctioned by the organization.

Via dedicated APIs for each target application, CASBs focus on activity performed in the SaaS application itself such as applying deep data protection controls to prevent malicious files from being shared, preventing Phishing messages, protect against data loss in messages and files and secure direct chats and collaboration channels. For example, API-based security can stop a malicious executable from being uploaded and propagated in a Slack channel.

For the long tail of shadow SaaS, CASB applies inline security such as access control, data protection and threat prevention. Here, user activity can be inspected to enforce policy related to granular application control, preventing threats, visibility into personal and unsanctioned apps and data loss prevention.

SSPMs focus on App Security Posture and Remediation

SSPMs differ from CASB solutions in that they assess and manage the security posture of an organization's SaaS applications.

To reduce an organization's SaaS attack surface, SSPMs uncover SaaS application settings that make an organization vulnerable and should be addressed. Examples include overly scoped access permissions, unused credentials, SaaS misconfigurations and weak native security settings like outdated encryption algorithms and insufficient authentication controls, to name a few.

SSPMs also require API-based integration and therefore can only manage security posture and remediate gaps for a limited number of SaaS applications.

Essentially, SSPMs focus on gap remediation within specific SaaS applications.

Uncovering SaaS-to-SaaS Connections

How well can CASBs and SSPMs discover shadow SaaS, including APIs, plugins and services?

CASBs can perform discovery via email notifications ("Welcome, you've signed up for a new SaaS service.") and by parsing logs of on-prem gateways or security service edges (SSEs), the latter which usually requires days-long integration work.

SSPMs on the other hand rely on discovering apps used in an ecosystem that are connected to sanctioned enterprise applications.

The bottom line is that SSPMs do uncover shadow SaaS connections, while CASBs are only designed to see either sanctioned connections, or see a long tail of shadow SaaS apps but mainly look at them in an isolated way that does not focus on SaaS-to-SaaS connections.

Preventing Risky SaaS-to-SaaS Connections in Real Time

Can CASBs or SSEs see when a SaaS tool's API token has been breached, and stop an attacker from exploiting it to read the emails, calendars, files and directories hosted on your SaaS platform? The answer is no.

And can SSPMs block malicious or risky SaaS interconnections (app-to-app connections) to prevent a breach in real time? While they do uncover SaaS to SaaS connections, most of them do not stop suspicious connections in real time, once again putting organizations at risk of supply chain attacks, account takeover and data theft.

The table below summarizes the differences between SSPM and CASB capabilities.

Capability	SSPM	CASB
Shadow IT Discovery	🔴	🟠
Posture Management	🟢	🔴
Uncover SaaS-to-SaaS Connections	🟢	🔴
Security long tail of SaaS apps (inline security)	🔴	🟢
Security sanctioned SaaS apps (API-based security)	🟢	🟢
Prevent risky SaaS-to-SaaS connections in real time	🔴 *	🔴

***The vast majority of SSPMs do not prevent Risky SaaS-to-SaaS connections in real time. More on this in the next few pages.**

While SSPMs and CASBs complement each other and considerably improve SaaS security, the above blind spots make it challenging for a CISO to maintain comprehensive oversight and control over the organization's entire SaaS landscape.

So how can the current situation be fixed?

SAAS SECURITY MUST-HAVES - THE ULTIMATE CHECKLIST

As shown above, SSEs and SSPMs complement each other, yet they lack a missing piece of the SaaS security puzzle as they do not monitor SaaS-to-SaaS connections, learn their behavior and risk-related attributes and most importantly—they do not stop risky SaaS connections in real time to prevent a breach from happening.

Here are several key capabilities to evaluate when choosing a SaaS security solution to finally complete the SaaS security puzzle.

Rapid SaaS Application Discovery

The traditional method of uncovering shadow SaaS involves lengthy integration work with CASBs or SSEs, which in turn analyze firewall or secure web gateway (SWG) logs of outbound traffic to discover unsanctioned SaaS applications. Other methods, as mentioned above, include discovery via email notifications, through endpoint-focused security or discovery that relies on directly integrating with a key SaaS platform.

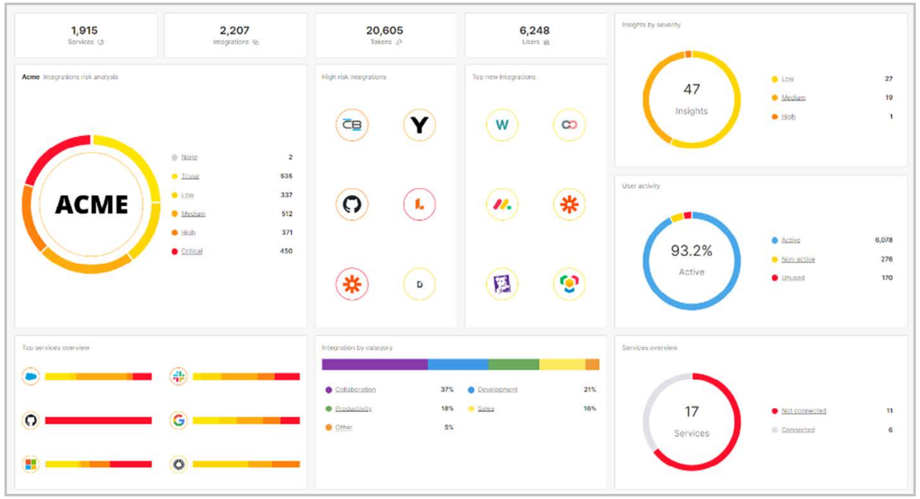
To be sure, there is no bulletproof way to ensure 100% visibility into all the SaaS applications used by every user in the organization. For example, an employee may download a SaaS plugin outside the corporate firewall, so the enterprise has no visibility into that application as no firewall logs will be created for that download.

Nonetheless, the more application discovery methods offered by a solution, the better your security posture will be.

By natively integrating with key SaaS platforms (normally via API), SaaS discovery is made faster, and can provide insights in minutes. A solution that also supports discovery via email and gateway logs will provide enhanced visibility and minimize blind spots.

What about application insights?

Look for a solution that uncovers useful information about every SaaS service, application or plugin used in your environment, such as a description of the service, compliance certifications that service has, location of operation of the service and reputation of the developer. A standardized risk score should be generated for each service based on the above information.



Prefer rapid SaaS discovery and standardized insights

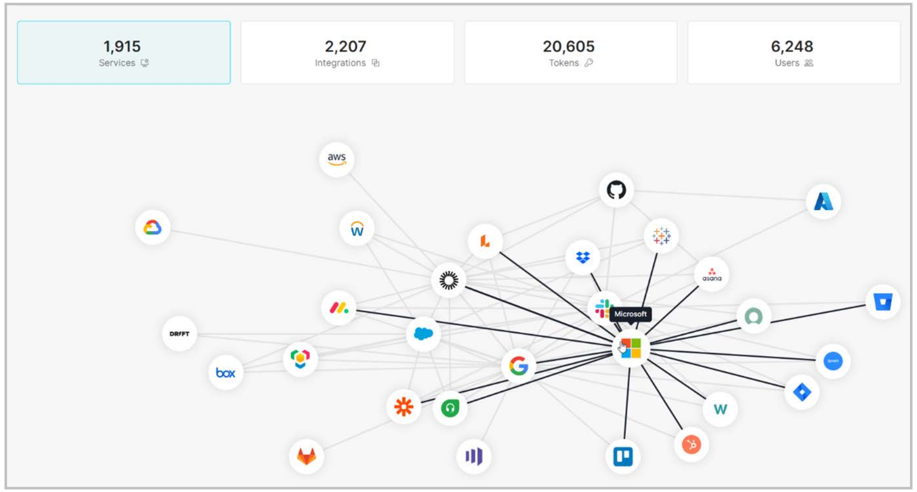
Visibility into SaaS-to-SaaS connections

Discovery of standalone apps is critical, but in itself, not enough.

Take a scenario where an employee uses a SaaS-based integration app that offers thousands of ready-made integrations to connect between almost any two SaaS services, for example between a marketing automation and a video distribution platfrom, or between an email exchange service and an application that makes it easy to manage email signatures across an organization. If any of these plugins or APIs is poorly secured, becomes breaches, deprecated or abandoned, those plugins and APIs open a backdoor to the data that resides in the enterprise SaaS they're connected to.

Seeing these connections is the first step to protecting them against abuse, exploitation or unintended exposure.

Think of this as the unknown unknowns, where the unknown integration points of SaaS-to-SaaS connections act as potential gateways for unforeseen security threats, making them critical blind spots in your organization's security posture.



Ensure Visibility into SaaS-to-SaaS Connections

Simplified Remediation of Configuration Drift

Identifying SaaS security gaps and configuration drift is central to continually reducing your SaaS attack surface, improving your security posture and maintaining regulatory compliance.

Look for a solution that can uncover, and offer simple remediation, for the scenarios below. Ideally, the solution should provide a standardized risk score across applications and prioritize remediation according to the level of risk.

Find applications that require permissions they don't use and revoke those permissions (zero trust for app-to-app communication)

- Example - An organization only uses some features of a SaaS service, and thus does not need all the permissions the service requires

Find malicious, breached, abandoned, legacy and deprecated services in your environment

- Example 1 – A user starts using an application without IT or security operations' approval, but that app is not in compliance or outright malicious
- Example 2 – A company used a sound SaaS service for several years, but the company developing the service has been out of business for a while, or discontinued the specific service, while the integration is still in place.

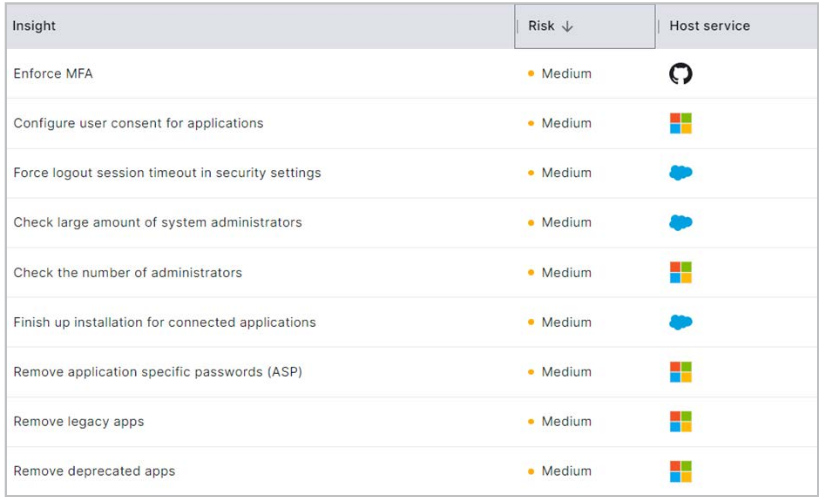
Find and fix configuration inefficiencies and configuration drift

- Example 1 – Native security settings - A feature that clearly needs to be enabled, is not enabled, such as enforcing multi-factor authentication or using a stronger encryption protocol.
- Example 2 – Excessive permissions - A new feature that was added to a SaaS service should be disabled, while it was enabled by default, for example, excessive read/write permissions.
- Example 3 – Inadvertent configuration changes - A change to a configuration setting affected another configuration setting, and now requires further actions to be taken. Case in point, in a well-known enterprise SaaS platfrom, some changes will disable the "Security Defaults" feature, and thus require the user to select and reactivate all the security defaults manually.
- Example 4 – Overly scoped identity permissions - Some configurations may not be additive in terms of security posture. Having guardrails that will let the user know if the additional configuration reduces security. For instance, it's ok to have up to four (4) Super Admins, but not more.

Find unused SaaS services, stale API keys or tokens, and stale users in those services and disable them.

- Example 1 – Out of 20 users of a SaaS service, only five (5) are active in the last 3 months, suggesting that there may be too many licensesand unused credentials
- Example 2 – An API key has not bee used in the past seven weeks,calling into question whether that service is even being used, or if it's no longer being used and the API token should be removed toreduce exposure
- Example 3 – Several employees decided to integrate an enterprise platform with a third party SaaS service. Uncover and eliminate redundant API clients used to integrate your organization with the same service
- Example 4 - Stale user accounts with active token, or even suspended accounts with active tokens, pose a significant security risk. In this scenario, the user no longer logs in to the account, yet there are APIs running on their behalf. A good example for this are apps like "SMS to Slack" (https://slack.com/apps/A044EAH46F2) , which can be installed on a specific Slack channel to get a text (SMS) message each time a message is sent on that channel. After leaving a company and being removed from the channel, the app stays installed, and keeps updating the user by text each time a message is sent on the channel

A SaaS security solution should be able to identify the above weak settings and configuration drifts as well as risky services and credentials that may serve as potential entry points, and provide single click remediation to fix them.



Insight	Risk ↓	Host service
Enforce MFA	● Medium	●
Configure user consent for applications	● Medium	⊞
Force logout session timeout in security settings	● Medium	☁
Check large amount of system administrators	● Medium	☁
Check the number of administrators	● Medium	⊞
Finish up installation for connected applications	● Medium	☁
Remove application specific passwords (ASP)	● Medium	⊞
Remove legacy apps	● Medium	⊞
Remove deprecated apps	● Medium	⊞

Seek a Solution that can Identify Security Gaps and Prioritize their Remediation

Behavior-based Threat Prevention

Detecting suspicious behavior is the cornerstone of real time threat prevention. Machine learning leverages historic data based on SaaS activity to proactively prevent account takeover, supply chain attacks, and data leakage.

By using Machine Learning to learn the baseline behavior of different SaaS services, and the interaction between different actors in those services, the SaaS security solution should be able to alert on, and automatically prevent, any anomalous actions for those services.

For best coverage, the baseline behavior should be established using data both from within an organization as well as across organizations. This means looking at how a SaaS service has functioned historically within the organization in which it's deployed, as well as across organizations. For example, if a SaaS service behaves one way in 100 different organizations, but another way in another in your organization, then this may be flagged as an anomaly.

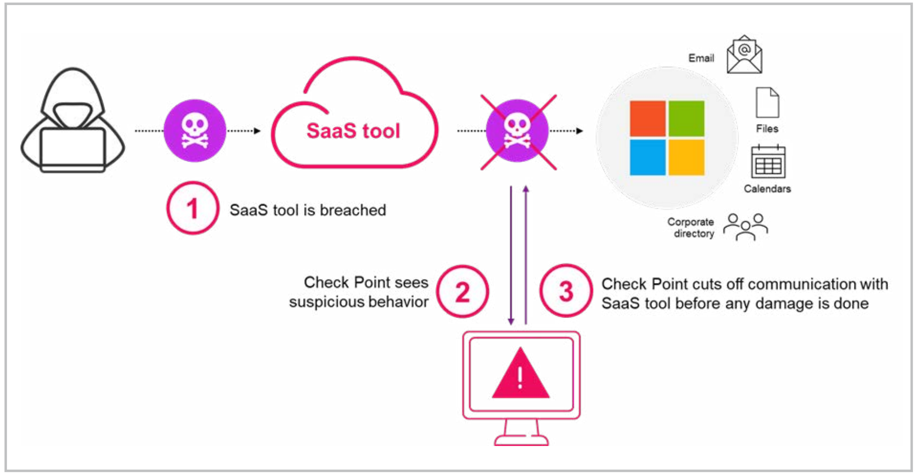
Some examples of behavior threat indicators include:

- Access attempts from an unfamiliar IP address
- Access attempt using an unusual call method
- Access attempts at odd hours
- Sudden and unauthorized elevation of user privileges
- High rate of failed login attempts
- Suspicious application requests
- And others

How it works

In the example below, machine learning is used to identify anomalous behavior and prevent a breach:

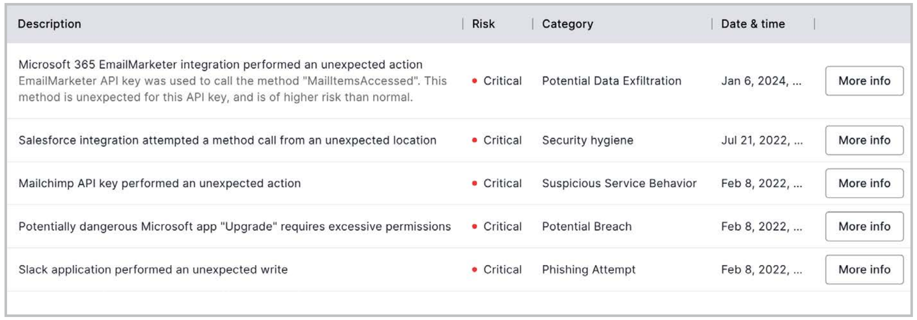
- A third party SaaS tool is connected to an enterprise SaaS platform.
- The SaaS tool is breached, enabling the threat actor to access sensitive data in enterprise platform, including emails, directory, calendars and files.
- A suspicious access attempt is identified based on user access from a new country, at an unusual time of day using a never-before-seen-here API calling method
- The API token is removed from the SaaS tool, preventing further activity by the threat actor in the target SaaS platfrom, such as exfiltrating data, sending spear-phishing emails or attempting to take over an account through socially engineered requests to the company's help desk services.


SaaS tool
Email
Files
Calendars
Corporate directory
1
SaaS tool is breached
Check Point sees suspicious behavior
2
3
Check Point cuts off communication with SaaS tool before any damage is done
!

Machine Learning is Essential for Automatically Preventing SaaS-based Threats in Real Time

While some organizations may opt for full automation, so that anomalous behavior is immediately stopped in real time to minimize potential damage, others may prefer a different approach where suspicious activity generates an alert and requires manual approval to remediate. And yet others may want to take an even more hands-off approach, and generate alerts that are read-only.

Whatever approach is taken, having alerts based on context provides a CISO and their team an advantage when it comes to prioritizing and mitigating threats. Particularly, the threats that have the highest impact on business-related KPIs.



Description	Risk	Category	Date & time	
Microsoft 365 EmailMarketer integration performed an unexpected action EmailMarketer API key was used to call the method "MailItemsAccessed". This method is unexpected for this API key, and is of higher risk than normal.	● Critical	Potential Data Exfiltration	Jan 6, 2024, ...	More info
Salesforce integration attempted a method call from an unexpected location	● Critical	Security hygiene	Jul 21, 2022, ...	More info
Mailchimp API key performed an unexpected action	● Critical	Suspicious Service Behavior	Feb 8, 2022, ...	More info
Potentially dangerous Microsoft app "Upgrade" requires excessive permissions	● Critical	Potential Breach	Feb 8, 2022, ...	More info
Slack application performed an unexpected write	● Critical	Phishing Attempt	Feb 8, 2022, ...	More info

Evaluate the flexibility of a solution to provide fully automated or approval-based prevention

Attribute-based SaaS Risk Assessment

To complement behavior-based threat prevention, threat intelligence Indicators should also be leveraged to automatically identify the inherent risk associated with an application as part of its overall risk score.

Examples of attributes that a solution should evaluate include:

- Deprecated applications
- Abandoned applications
- Applications with expired compliance certifications
- Applications running unpatched, vulnerable versions of the service
- Application's source country
- User popularity as indicated by number of customers, peer review ratings etc.
- And other attributes that help identify risk

Intuitive Rollout and Management

To ensure quick time-to-value of your SaaS security solution, look for services that:

- SaaS-native ddeployment with no agents, hardware, or otherwise local presence required.
- Effortless rollout that can be completed in a couple of minutes and a few clicks.
- Availability of Information, insights and policies within minutes of installation
- Intuitive administration that lowers the barrier to managing saas security, so that no prior expertise required

Alerts on Changes affecting Regulatory Compliance

A SaaS security solution should ideally facilitate maintaining tight compliance with regulatory mandates, such as GDPR, PCI-DSS, ISO, SOC 2, SOX, HIPAA and others.

To this end, ensure your solution provides notifications and automated fixes when a configuration drift leads to compliance mistakes. Some examples of configuration mistakes that may lead to compliance drift are:

- Insufficient user authentication
- Weak encryption of data in transit
- Absence of user consent for certain applications

Additionally, seek a solution that can continuously validate your SaaS services to eliminate supply-chain compliance drift, for example:

- Continuously monitor the compliance certificates of your vendors, so you know if a vendor loses, or fails to renew a certificate such as SOC 2 compliance or ISO 27001
- Continuously monitor your vendors' commitments to you, including information about sub-processors, changes in privacy policy or terms of service, and data processing location of your data

A Zero Trust Approach to App Access

While companies are familiar with the concept of Zero Trust when it comes to user access, the concept of zero trust application access is not widely understood or implemented.

Businesses are increasingly following the principle of "always verify, never trust," and investing in deploying systems that ensure least-privilege access controls are enforced when users access company resources, such as datacenters, IaaS, SaaS and internal applications.

However, when it comes to application-to-application or SaaS-to-SaaS access, Zero Trust is rarely implemented. The result: SaaS tools, shadow IT, APIs and plugins have access to vast volumes of sensitive data which they do not actually require to function.

The capabilities mentioned above support taking this critical step to implement a true Zero Trust architecture in your enterprise across hundreds of SaaS applications.

Harmony
SaaS

HOW CHECK POINT HARMONY SAAS HELPS

Harmony SaaS is the most advanced solution for automatically preventing SaaS-based threats in real time. Using machine learning engines that look at app-to-app connections, Harmony SaaS protects your SaaS ecosystem against threats such as data theft, account takeover and supply chain attacks.

Unlike traditional solutions, Harmony SaaS installs within minutes, discovers your SaaS applications, analyzes security posture gaps, provides single-click remediation, and automatically prevents threats as they arise.

Take the guesswork out of SaaS security and compliance, starting today.

Book a demo with a Harmony SaaS expert, or **test drive it** for yourself and see why Harmony SaaS provide the best time-to-value for your SaaS security.

To learn more about Harmony SaaS, visit **checkpoint.com/harmony/saas/**

CHECK POINT™

About
Check Point Software Tchnologies Ltd.

Check Point Software Technologies Ltd. is a leading provider of cyber security solutions to corporate enterprises and governments globally. Check Point Infinity's portfolio of solutions protects enterprises and public organizations from 5th generation cyber-attacks with an industry leading catch rate of malware, ransomware and other threats. Infinity comprises three core pillars delivering uncompromised security and generation V threat prevention across enterprise environments: Check Point Harmony, for remote users; Check Point CloudGuard, to automatically secure clouds; and Check Point Quantum, to protect network perimeters and datacenters, all controlled by the industry's most comprehensive, intuitive unified security management; Check Point Horizon, a prevention-first security operations suite. Check Point protects over 100,000 organizations of all sizes.

To learn more about us, visit: www.checkpoint.com

Contact us

Worldwide Headquarters
5 Ha'Solelim Street, Tel Aviv
67897, Israel
Tel: 972-3-753-4555
Fax: 972-3-624-1100
Email: info@checkpoint.com

U.S. Headquarters
959 Skyway Road, Suite 300,
San Carlos, CA 94070
Tel: 800-429-439 / 650-628-2000
Fax: 650-654-4233