stratascale
CYBERSECURITY DIVISION OF SHI

Stratascale Cybersecurity Brief

When Trusted Systems Become the Attack Path

Volume 1 | First Quarter 2026

Table of

Contents

Executive Summary

Introducing the Stratascale Cybersecurity Brief, a quarterly report built for security leaders who need clarity, not noise.

This brief is designed to provide an executive-level view of how the cyber risk landscape is evolving and what those changes mean for organizational risk and decision-making in the months ahead.

Each issue is intended to close the gap between insight and action by offering practical guidance and real-world recommendations you can use, not surface-level commentary.

This quarter, our analysis of real-world incidents, active attack campaigns, and recent vulnerability disclosures points to a meaningful shift in how adversaries gain access.

While the specific techniques vary, the underlying pattern is consistent: **attackers are increasingly entering environments through systems and access paths that organizations already trust.**

Across the incidents examined, we observed that threat actors are relying less on traditional intrusion methods such as endpoint compromise, password spraying, and perimeter exploitation.

Instead, they are operating inside trusted control planes, including SaaS integrations, CI/CD pipelines, federated identity platforms, and non-human identities.

These systems are designed to automate access and enable scale.

As a result, they often provide persistent permissions, broad reach, and activity that appears legitimate in logs, making abuse difficult to detect and contain.

This shift has important implications for security leadership. When abuse occurs inside trusted systems, controls such as multifactor authentication, endpoint detection, and network defenses may offer limited protection because access is inherited through OAuth tokens, service accounts, or automation credentials.

Several of the incidents highlighted in this brief show that administrative actions, such as disabling an identity or revoking access, do not always translate into effective enforcement at runtime.

To help you interpret this shift and understand its impact on your organization, **the brief breaks down four recurring attack patterns across trusted systems.** For each pattern, we outline the core failure mode, highlight real-world signals and incidents that demonstrate how the issue manifests, and explain why it matters from a business and risk perspective.

Each section also includes targeted operational recommendations you can apply within your environment to reduce exposure related to that specific pattern. Dive in and get informed.

"Abusing trust is a more scalable attack strategy than breaking controls. It is harder to detect, easier to maintain, and often provides immediate access to what attackers want. By starting with trusted identities and systems, attackers inherit trust and access simultaneously."

Jordan Mauriello, CTO
SHI

Recurring Attack Patterns Across Trusted Systems

The incidents and vulnerability disclosures reviewed in this section show attackers shifting how they move through enterprise environments. Rather than forcing entry through overt compromise techniques, they increasingly advance by abusing trusted access paths that were intentionally designed to automate connectivity and scale operations.

This shift matters because these trust paths often:

Bypass MFA and interactive login controls, relying on OAuth tokens, service identities, and build credentials rather than user authentication.

Appear legitimate in logs, since activity is executed by approved applications, automation, or non-human identities.

Expand blast radius, where a single upstream compromise, such as a vendor or workflow, can impact many downstream systems or customers simultaneously.

To illustrate how abuse of trusted systems manifests in real environments, we highlight four attack patterns that share a common failure mode.

In each case, administrative intent is not reliably enforced at runtime, allowing delegated trust to become the attacker's most efficient and scalable path.

Each pattern below combines real-world signals, relevant advisories, attack behavior, and business impact to show how trusted access is exploited in practice.

Designed to be action-oriented, you can use the exposure signals to assess relevance, then apply the outcome-focused recommendations, ownership, and time horizon to guide prioritization.

You can read the patterns individually to address specific concerns, or review all four together to identify shared trust weaknesses that require coordinated action. This is important as trusted systems increasingly serve as the primary attack surface.

Overprivilege SaaS Integrations

SaaS Supply Chain Trust Abuse

Early in March, AppOmni reported that ShinyHunters (tracked as UNC6040) claimed a breach of Woflow, a SaaS integration provider, alleging the exfiltration of “hundreds of millions of records” and threatening public disclosure. AppOmni noted that extortion messaging referenced a March 6, 2026, deadline.[1]

Regardless of confirmation status, the case is used to illustrate a broader trend: attackers increasingly target integration-heavy SaaS vendors to inherit OAuth/API trust into downstream customers, enabling lateral movement and data access at scale.[2] [3]

Real World Signal
Salesforce OAuth Vishing Campaigns (UNC6040)

Attackers used voice phishing to coerce employees into authorizing malicious OAuth applications. No platform vulnerability was exploited.

Access was inherited through approved OAuth workflows and persisted across Salesforce and downstream SaaS platforms.This campaign illustrates how attackers increasingly “log in, not break in,” abusing delegated SaaS trust and non-human identities to obtain durable access that bypasses MFA and appears legitimate in logs.

Why This Matters

Modern SaaS ecosystems rely on OAuth permissions, API tokens, and non-human identities to connect business-critical apps (CRM, automation, analytics, AI tooling). When an attacker compromises an integration provider embedded across many tenants, they can potentially gain indirect access to multiple organizations at once.

Attack Pattern

High-Level Chain

1. Compromise an integration-rich SaaS vendor that holds durable OAuth/API access in customer environments.
2. Abuse OAuth access tokens / refresh tokens and inherited service permissions to perform legitimate-looking” API actions.
3. Move laterally across SaaS-to-SaaS connections (where MFA and network controls are not in-path).
4. Aggregate and exfiltrate sensitive datasets through API-level access.
5. Apply extortion pressure via public claims, deadlines, and staged releases (pattern described as a recurring extortion cadence).

Business Impact

- **Silent Data Exposure:** Access occurs through approved APIs, complicating detection and incident scoping.
- **Downstream Amplification:** A single vendor compromise can create a multi-customer blast radius and reputational contagion.
- **Extortion Leverage:** Attackers use public victim naming and deadlines to increase pressure and perceived consequences.

---

[1] AppOmni (ShinyHunters/Woflow)
[2] CyberInfos weekly report
[3] Microsoft News supply-chain reporting

Exposure Signals
Fast Self-Assessment

You are at higher risk if...

- Third-party integrations retain broad read/write scopes to core SaaS platforms.
- OAuth tokens are long-lived and not routinely rotated/revoked across applications.
- You cannot quickly answer: “Which vendors have durable API access to our CRM/HR/ticketing/data platforms, and what scopes do they hold?”

Recommendations
Outcomes, Not Runbooks

- **SaaS Integrations are Treated as Identities**: Every integration is inventoried, has a documented owner, is periodically reviewed, and is removed when no longer required.
- **Least Privilege is the Default:** Read-only access is used whenever possible. Write or administrative scopes require explicit business justification and regular recertification.
- **Token Risk is Managed**: Short lifetimes where feasible, revocation capability tested, and anomalous API behavior monitored.
- **Security Visibility Extends into SaaS Control Planes**: Monitoring is not limited to network, endpoint, or user login activity, because MFA/SSE/CASB alone will not observe SaaS-to-SaaS misuse.

Ownership
IAM, SaaS Platform Owners, and Security Engineering

Time Horizon
Now: Inventory and revoke unknown
30 Days: Least privilege and recertification
Quarterly: Continuous monitoring and governance

“Organizations should double down on identity security by enforcing phishing-resistant MFA, applying behavioral analytics, and adopting zero trust architectures to constrain attacker movement and limit blast radius.”

Casey Corcoran, Field CISO
Stratascale

CI/CD Still a Target

Mass-Exploitable via Automated/Agentic Discovery

On March 4, Orca reported on "HackerBot-Claw," an automated campaign that exploited misconfigured GitHub Actions to gain CI execution, steal privileged tokens, and, in severe cases, take control of repositories.[4]

Shortly after, Aqua Security confirmed a real-world impact of this technique when Trivy was compromised via GitHub Actions.[5] The attack disrupted the repository, removed releases, and published malicious artifacts after a pull request triggered CI workflows and enabled unauthorized API access using a compromised personal access token.[6]

Real World Signal

Trivy GitHub Actions "Tag Poisoning"

After an initial CI compromise, attackers force-pushed existing version tags to malicious commits. Pipelines referencing previously trusted versions continued executing poisoned code even after remediation steps were taken, demonstrating how CI trust artifacts can outlive containment.

This incident shows how administrative intent in CI/CD environments does not reliably translate into runtime enforcement. Once automation and credentials are compromised, attackers can sustain impact at machine speed.

Why This Matters

CI/CD is not "developer tooling." It is a production trust boundary where code becomes artifacts, releases, and deployed software. When workflows are misconfigured, automation can become a credential oracle, granting attackers the write-level identities needed to modify source, releases, or downstream distribution channels.

Attack Pattern

High-Level Chain

Recurring Exploitation Conditions Observed by Orca

- Workflows triggered on PR events (especially patterns involving pull_request_target) running with elevated permissions and/or secrets.
- Untrusted input (branch names, filenames, public repository content) influencing execution or checkout behavior, enabling arbitrary command execution or token leakage.

Campaign operational loop:

1. Scan public repos for vulnerable workflow patterns.
2. Open public repositories crafted to trigger CI execution in privileged contexts.
3. Achieve CI execution or secret/token exposure.
4. Use stolen tokens (GITHUB_TOKEN or PAT) to push commits, delete releases, publish malicious artifacts, or modify repository settings.

"When automation is compromised, it doesn't probe slowly. It executes at machine speed, often completing its objective before defenders can react."

Rob Forbes, Field CISO
Stratascale

[4] Orca Security HackerBot-Claw deep dive
[5] Aqua Security Trivy incident
[6] GitHub pull_request_target security model documentation

Confirmed Business Impact Example (Trivy)

Aqua's incident notes impacts include the repository being made private/ renamed, releases being deleted (0.27.0–0.69.1), and a malicious artifact being created/published for a VS Code extension; their follow-up timeline includes a CI-triggering PR and subsequent unauthorized API activity using a compromised PAT.

Exposure Signals

Fast Self-Assessment

You are at higher risk if:

- You use pull_request_target and workflows can be influenced by untrusted PR code or user-controlled inputs.
- CI tokens/PATs have write permissions broader than required.
- Workflows fetch/execute external scripts or run PR-supplied code without strong isolation and validation.

Recommendations

Outcomes, Not Runbooks

What "good" looks like for CI trust and credential control:

- **CI Tokens Default to Read Only:** Write permissions are granted only when a workflow explicitly requires them, with scopes narrowly constrained to the job and function.
- **pull_request_target Usage is Intentional and Constrained:** The trigger is used only when required, and workflows are designed so untrusted code or inputs cannot influence execution paths that access secrets or elevated credentials.
- **Workflow Changes are Treated as High-Risk Assets:** CI workflows are reviewed with the same rigor as production code, and legacy branches or workflows are not permitted to remain in a vulnerable state.
- **Release and Artifact Pipelines Limit Blast Radius:** Guardrails are in place to reduce destructive impact, including credential rotation readiness and immutable or protected release practices where supported.

Ownership

AppSec, DevOps, and Engineering Leadership

Time Horizon

Now: Permissions and trigger review
30 Days: Guardrails and automation
Quarterly: Release integrity and continuous CI posture checks

Disable Doesn’t Mean Dead

Authentication Can Proceed via a Disabled IdP

CVE-2026-3009 is a Keycloak flaw where the IdentityBrokerService.perform Login endpoint can allow authentication flows to proceed for an IdP even after it has been disabled by an administrator.

An attacker who knows the IdP alias can reuse a previously initiated login request/session parameters to bypass the administrative restriction, undermining emergency containment measures.[7][8][9]

Real World Signal

Fortinet FortiCloud SSO Trust Boundary Collapse

Attackers exploited FortiCloud SSO trust relationships to gain unauthorized access across separately owned devices. Emergency mitigations required Fortinet to temporarily disable FortiCloud SSO entirely. This is a clear example of delegated identity trust collapsing at the control plane. When the trusted SSO path becomes the attack path, perimeter and user-auth controls provide no protection.

Relevant Vulnerabilities

Fortinet FortiCloud SSO Authentication Bypass (CVE-2026-24858)

Attackers exploited FortiCloud Single Sign-On to authenticate into customer environments they did not own by abusing legitimate SaaS-backed SSO trust. Fortinet was forced to temporarily disable FortiCloud SSO globally to contain active exploitation.

This is a clear example of delegated identity trust collapsing at the control plane, where trusted SSO paths become the attack path, and administrative controls fail to prevent runtime abuse.

Keycloak Broker Bypass (CVE-2026-3009)

This vulnerability allows authentication flows to proceed even after an IdP has been administratively disabled, undermining the assumption that disablement functions as a reliable containment control.

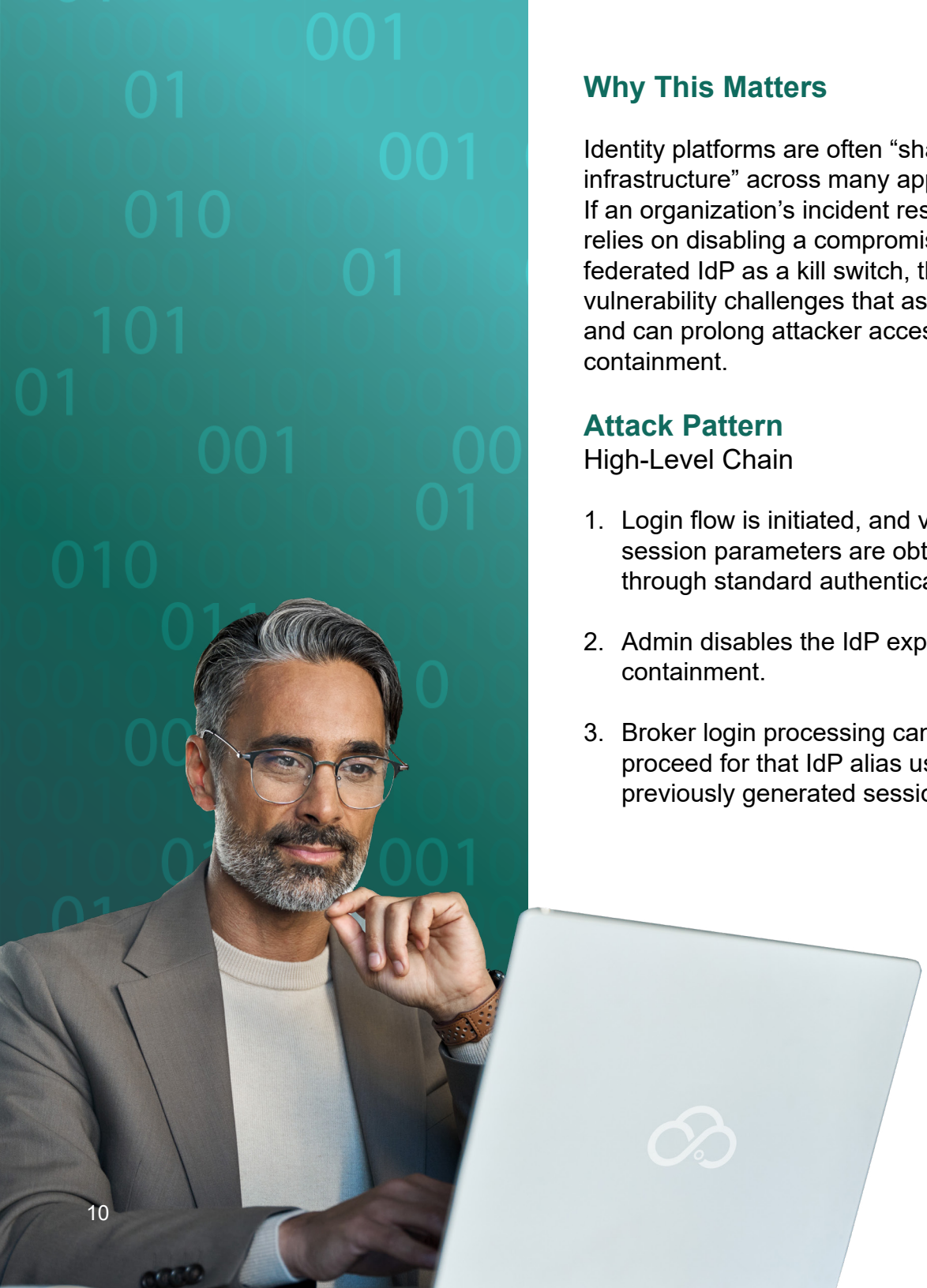
It demonstrates that administrative intent is not reliably enforced at runtime. During incident response, attackers may retain access through residual session artifacts, extending dwell time.

Cisco Catalyst SD-WAN Controller Authentication Bypass (CVE-2026-20127)

A critical authentication bypass in Cisco Catalyst SD-WAN Controller (vSmart) and SD-WAN Manager (vManage) lets an unauthenticated remote attacker bypass peering authentication and obtain high-privileged administrative access, enabling NETCONF-based manipulation of SD-WAN fabric. Configuration demonstrates that control-plane authentication failures can invalidate containment assumptions at scale, where compromise of trusted infrastructure enables systemic impact across environments.

> “Disabling or revoking access does not guarantee safety. Security requires the full lifecycle: approving access, monitoring it, maintaining it, and validating that it is actually removed when it is no longer needed.”
>
> **Jordan Mauriello, CTO**
> SHI

[7] NVD CVE-2026-3009
[8] Keycloak issue tracker
[9] Red Hat advisory



Why This Matters

Identity platforms are often “shared infrastructure” across many applications. If an organization’s incident response relies on disabling a compromised federated IdP as a kill switch, this vulnerability challenges that assumption and can prolong attacker access during containment.

Attack Pattern

High-Level Chain

1. Login flow is initiated, and valid session parameters are obtained through standard authentication steps.
2. Admin disables the IdP expecting containment.
3. Broker login processing can still proceed for that IdP alias using the previously generated session artifacts.

Business Impact

- Containment failure: “disable IdP” may not stop broker authentication immediately.
- Unauthorized access continuity across Keycloak-reliant applications, increasing dwell time and incident scope.

Recommendations

Outcome-Based

- **Patch to vendor-fixed releases per advisories.** Treat as a priority issue because the vulnerability bypasses an explicit administrative security control.
- **Update incident playbooks so “Disable IdP”** includes validation steps to confirm all authentication paths are fully blocked, reducing the risk of false containment confidence.

Ownership

IAM and Security Architecture

Time Horizon

Now: Patch planning and containment validation
30 Days: Playbook updates
Quarterly: Automated verification controls

High-Impact Identity Bypass

Disabled IdP Still Issues Tokens

CVE-2026-1486 is a Keycloak logic flaw in the jwt-authorization-grant flow where the server fails to verify whether an IdP is enabled before issuing tokens. If an attacker possesses a disabled IdP's signing key, they can craft valid JWT assertions that Keycloak will accept and exchange for access tokens, despite the IdP being administratively disabled.[10] [11] [12]

Relevant Vulnerability

Keycloak JWT Authorization Grant Bypass (CVE-2026-1486)

This flaw allows token issuance from a disabled IdP if signing keys are retained, enabling attackers to mint valid tokens without interactive authentication. Cryptographic trust artifacts persist beyond the trust decision itself. Disablement without key lifecycle governance creates a false sense of containment.

Why This Matters

This is a textbook "trust artifact outlives trust decision" problem: organizations disable IdPs during offboarding or compromise response, but if signing keys are retained/compromised and the platform fails to enforce "disabled," the attacker can continue to mint access through token flows that look cryptographically valid.

Attack Pattern

High-Level Chain

1. The attacker obtains a disabled IdP's signing key material (or retains it after compromise/offboarding).
2. The attacker crafts a JWT assertion with an issuer matching the disabled IdP configuration and a valid signature.
3. Keycloak processes the assertion via issuer lookup that does not exclude disabled IdPs and issues access tokens.

Business Impact

- Unauthorized token issuance enabling access to protected resources without normal interactive login patterns.
- Containment weakness if disabling an IdP is treated as sufficient without key rotation/revocation and token flow validation.

Recommendations

Outcome-Based

- **Patch keycloak** to vendor-fixed versions per advisories.
- **Treat IdP disablement as an offboarding action** that requires full key lifecycle controls, including rotation and revocation of signing keys and validation of token issuance paths.
- **Audit disabled IdPs** for residual trust artifacts and ensure the "disabled" state cannot be used to mint valid tokens.

Ownership

IAM and Security Architecture

Time Horizon

Now: Patch and audit
30 Days: Key governance and playbook updates
Quarterly: Automated federation offboarding controls

---

[10] NVD CVE-2026-1486
[11] GitHub Security Advisory
[12] SentinelOne analysis

The Key Takeaway

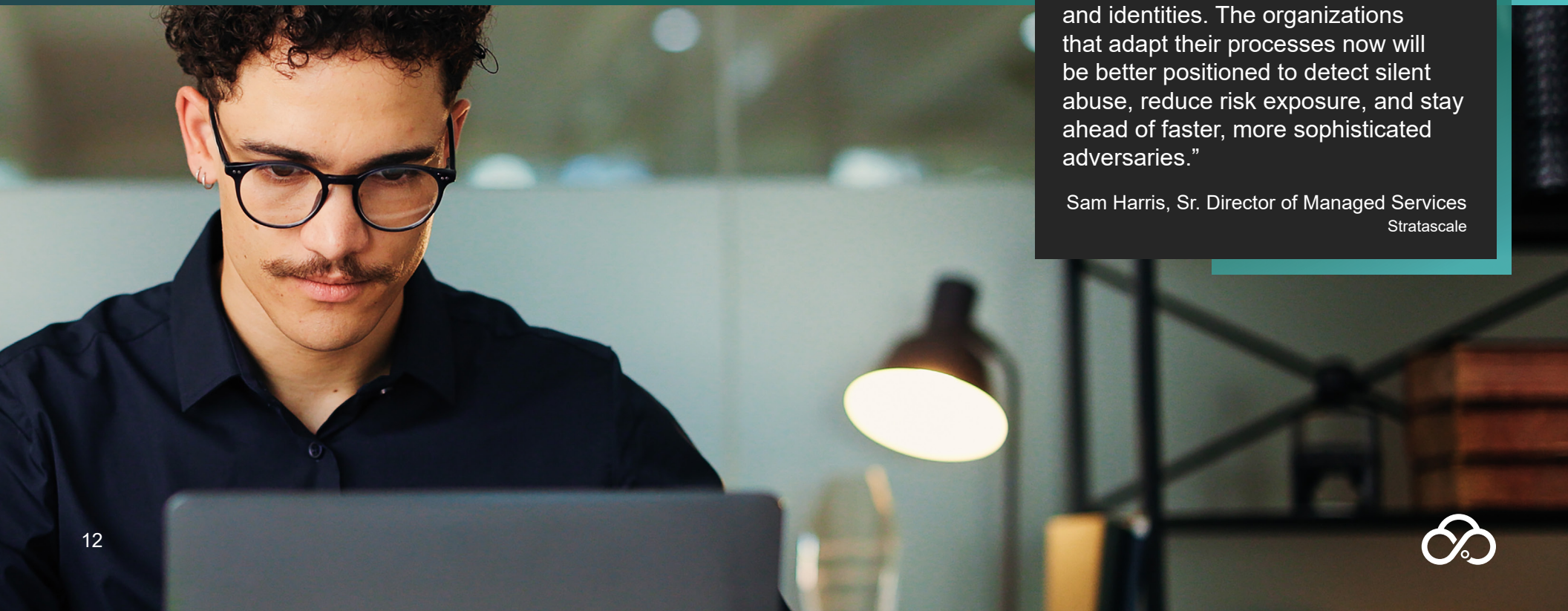
The four patterns in this brief point to a consistent reality. **Today's most damaging access paths often do not look like intrusion; they look like normal operations.** Attackers are increasingly exploiting delegated trust in SaaS integrations, automation pipelines, and identity systems, and in several cases, administrative intent does not translate into reliable enforcement at runtime.

The result is a control gap where organizations believe access has been restricted, but trust artifacts, tokens, workflows, or federation paths can still allow activity to continue.

The takeaway is to treat trusted systems as primary control planes and to validate that security decisions are enforceable in practice. That means tightening governance over non-human access, reducing overprivileged integrations and automation, and verifying that disablement, revocation, and containment actions actually terminate access across all relevant paths.

"Modern attacks no longer rely on breaking controls. They exploit trust, identity, and legitimate access that organizations assume is safe. Disabling accounts, patching systems, or deploying more tools is no longer enough. Security leaders must validate that intent translates into enforcement, ensure access is truly revoked, and continuously govern SaaS integrations and identities. The organizations that adapt their processes now will be better positioned to detect silent abuse, reduce risk exposure, and stay ahead of faster, more sophisticated adversaries."

Sam Harris, Sr. Director of Managed Services
Stratascale



Executive-Ready Action Plan

The recommendations within each attack pattern in this brief are purpose-built for that specific scenario and focus on operational actions your team can execute. The action plan below elevates those insights into a strategic, theme-driven set of priorities that reduce exposure across all four patterns.

Use this list to align with your team and turn these priorities into an action-oriented strategy you can act on today.

Immediate | 0–14 days
Stop the Obvious Trust Leakage

- Inventory and review third-party SaaS integrations with write or broad scopes; disable unowned or unused integrations.
- Audit GitHub Actions for high-risk patterns (for example, pull_request_ target combined with secrets and untrusted inputs); enforce least-privilege token permissions.
- Identify Keycloak deployments and validate patch posture for CVE-2026-1486 and CVE-2026-3009 where applicable.

Near Term | 15–45 days
Make Containment Real

- Update incident response playbooks so “disable” actions include verification steps, such as IdP disablement validation, token issuance checks, and session invalidation.
- Implement recurring recertification for SaaS apps and OAuth scopes, with enforced ownership for non-human identities.
- Add CI/CD guardrails, including hardened triggers, input handling discipline, and automated checks for dangerous workflow combinations.

Quarterly | 45–90+ days
Operationalize Trust Governance

- Build continuous monitoring for SaaS token misuse and anomalous API behavior across critical SaaS platforms.
- Establish CI posture management as a control domain (not ad-hoc YAML reviews), with continuous scanning and policy enforcement.
- Formalize federation offboarding with key rotation/revocation workflows and automated validation to prevent “ghost trust” paths.

stratascale
CYBERSECURITY DIVISION OF SHI

You Partner in Success

For additional guidance or support in applying the insights from this brief, **click here to contact Stratascale.**

Our team can help assess trust paths, validate enforcement controls, and align next steps across your organization.

stratascale.com