ConductorOne

A Practical Approach to Achieving Zero Standing Privileges (ZSP)

Standing privileges—whether forgotten, unused, or simply overprovisioned—create an ever-widening attack surface that cybercriminals are all too eager to exploit. Yet many organizations hesitate to implement zero standing privileges (ZSP), concerned about disrupting workflows or overwhelming their IT teams.

Learn why ZSP is an effective approach to lowering risk and how to implement it without creating friction or slowing productivity.

What is zero standing privileges (ZSP)?

Zero standing privileges (ZSP) is a security model that enforces the principle of "no default access" by granting user accounts, systems, or applications only the minimum levels of access necessary to complete specific tasks—and only for a limited time.

While traditional cybersecurity strategies emphasize external defenses, ZSP's foundation is based on a "zero trust" security architecture.

Instead of assuming that certain users can be permanently trusted with high-level access (standing privileges), zero trust advocates for a "trust no one, verify everything" approach to access. This means every request for privileged access, regardless of who makes it, must be verified, validated, and time-limited.

Benefits of zero standing privileges

Mitigates insider threats

ZSP minimizes the risk posed by both malicious and negligent insiders by enforcing strict, temporary access controls.

By eliminating standing privileges, companies ensure that insiders do not retain unnecessary access to sensitive data. This significantly reduces the potential for privilege abuse—a common risk with administrative accounts that retain ongoing permissions.

When administrative access is needed, ZSP ensures that it is granted only temporarily, and precisely for the required task.

This targeted access model prevents unauthorized tampering or data leaks, creating a secure environment where employees are granted only the specific privileges required to complete a given task.

Controls unauthorized access and reduces attack surfaces

ZSP limits unauthorized access opportunities through just-in-time (JIT) access and just-enough access (JEA) policies. With these measures, access rights are granted strictly on an as-needed basis, reducing exposure and minimizing potential entry points that hackers or other malicious actors might target.

By tightly controlling access rights, ZSP creates a more secure access environment that leaves little room for vulnerabilities in the system. This approach is particularly effective in defending against cyberattacks and preventing data breaches.

Should an attacker obtain user credentials, the absence of standing privileges makes it far more challenging to escalate permissions or gain deeper access to critical systems, thereby reducing the organization's attack surface and limiting the impact of unauthorized access.

Ensures regulatory compliance and lower costs

Regulations like GDPR, CCPA, and HIPAA require strict data access controls to safeguard personal and sensitive information.

ZSP aligns directly with these requirements by enforcing controlled and temporary access, ensuring that instead of retaining standing access, users only access data when necessary.

Additionally, because enforcing ZSP requires continuous monitoring and logging of access events, it creates comprehensive audit trails to demonstrate compliance. These audit trails provide the transparency and accountability necessary to satisfy regulatory requirements, streamlining audits and helping organizations avoid penalties.

Furthermore, ZSP can positively impact cyber insurance premiums. Since ZSP significantly reduces the risk of privilege-based incidents, insurers may view organizations with ZSP as lower risk, potentially leading to lower premiums and more favorable coverage terms.

Principles of zero standing privileges

ZSP operates on four foundational principles designed to enforce stringent access controls and support identity security: zero access by default, just-in-time (JIT) Access, just-enough access (JEA), and automated privilege revocation.

Zero access by default

In a ZSP framework, every user, application, and process begins with zero access to sensitive systems, data, and resources. This zero-access starting point limits exposure, ensuring that no entity has access beyond its immediate, defined need.

How it works:

→ Users and systems start with zero standing privileges, creating a secure baseline.

→ Each access request is validated and scoped to the specific task, ensuring compliance with least privilege access principles.

→ Requests are evaluated based on predefined security policies and risk factors, such as data sensitivity and user context (e.g., location, device, and behavior).

Did you know? "Zero access by default" is an extended form of the principle of least privilege (POLP), which dictates that users, applications, and systems should have only the minimal level of access necessary to perform their specific tasks, and nothing more.

In a ZSP context, this principle is taken a step further by removing all standing privileges, so access is only provided when required.

Recommended → Least Privilege Access vs. Zero Trust

Just-in-time (JIT) access

JIT access is a dynamic access control method that grants users temporary access to systems or resources only when needed for a specific task. Access is granted at the moment of need and removed immediately afterward.

How it works:

→ Access is requested at the moment it's required, either manually (through a request process) or automatically (based on context-driven triggers).

→ Access privileges are limited by time, usually with an expiration period preset by security policies.

→ Multi-factor authentication (MFA) and other forms of contextual validation, such as device checks, are typically required before granting access.

Just-enough access (JEA)

JEA is a granular access control method that ensures users receive only the exact privileges necessary to complete a specific task—no more, no less. This minimizes the risk of privileged accounts gaining unauthorized access to unrelated systems.

How it works:

→ Permissions are scoped down to the minimum needed for the task at hand, often by segmenting roles and functions within the system.

→ Access control policies are highly specific, preventing users from viewing, editing, or interacting with data beyond the task's requirements.

→ Role-based access control (RBAC) and attribute-based access control (ABAC) models can assist in enforcing JEA by aligning permissions with predefined task roles or attributes (such as department, seniority, or task specifics).

Automated privilege revocation

After a task is completed, all privileges are automatically revoked, returning the user or system to a "zero-access" state. This way, there's a secure access control and provisioning system.

How it works:

→ Access management systems automatically track when a privilege is granted and set expiration timers or conditions for revocation.

→ Privilege revocation triggers can be based on specific events, such as the end of a session or the completion of a task.

→ Privileges may also be revoked proactively if unusual behavior or activity is detected, aligning with risk-based access controls.

Practical example: How ZSP works

💡 **Scenario →** A DevOps engineer needs temporary access to a cloud-based SaaS application for troubleshooting and uses SSH to connect to a remote server.

Phase 1: Zero access by default

→ The DevOps engineer has no standing access to the SaaS application or SSH connections by default, reducing the risk of unauthorized access.

Phase 2: Requesting just-in-time (JIT) access

→ When access is required, the engineer submits a request, which is verified based on context, such as device and location, ensuring secure remote access through SSH and temporary permissions for the SaaS application.

Phase 3: Just-enough access (JEA)

→ The engineer receives only the permissions necessary to complete the troubleshooting task—no more. This limits visibility and control within the SaaS application and ensures SSH access is confined to the specific server needed.

Phase 4: Automated privilege revocation

→ Once the task is complete, the system automatically revokes the engineer's permissions, restoring zero-access status. The removal of standing access guarantees that no lingering permissions remain in the SaaS application or the SSH connection, maintaining strict control over privileged access.

Step-by-step guide:
How to implement zero standing privileges

Step 1: Conduct a privileged access assessment

Objective:
Begin by evaluating your organization's current privilege structure to identify who has access to what resources, and to what extent. A privileged access assessment will reveal gaps, excessive privileges, and areas of potential security risk.

Process:

→ Conduct an inventory of all users, systems, and applications with access to sensitive data or privileged systems. This inventory helps establish a baseline for Identity Governance and Administration (IGA) standards by ensuring each role's privileges align with its actual access needs.

→ Analyze the list to determine which users have unnecessary or overly broad access. Excessive privileges can occur due to role changes, onboarding/offboarding issues, or improperly defined roles.

→ Group users and systems by risk level based on their roles and potential impact if compromised.

- For example, IT administrators, who often need elevated access, would be considered higher risk, while general users may pose a lower risk.

💡 **Pro Tip →** Consider using Identity and Access Management (IAM) tools like ConductorOne to conduct automated privilege audits. This helps identify privilege gaps, manage roles, and generate reports.

Related → IGA vs. PAM

Step 2: Create an access control structure

Objective:
Define a clear access control structure using RBAC and ABAC to restrict user permissions based on job roles or specific attributes, ensuring each user has the least privilege necessary to perform their duties.

Process:

→ For RBAC, categorize users into roles based on their responsibilities and assign each role specific access permissions.

- For example, IT support staff might need limited access to server logs, while HR might only have access to personnel files.

→ For ABAC, establish attributes (such as location, department, and time of access) that control access dynamically.

- For instance, you could allow access to certain files only during business hours and restrict access outside of those hours.

Alternatively, you can use a combination of RBAC and ABAC. Let RBAC handle the broader access control framework and ABAC add conditional restrictions based on attributes. This hybrid approach allows for granular and flexible access control.

Recommended → 7 Principles for Least Privilege Access Implementation

Step 3: Restrict access to an on-demand basis

Objective:
Implement JIT access mechanisms to provide time-limited access to resources, ensuring users do not retain standing privileges and only have access when necessary.

Process:

→ Define policies for granting temporary access. JIT access is typically requested via an identity governance and administration (IGA) or privileged access management (PAM) solution and requires multi-factor authentication (MFA) for verification. Define the maximum time allowed for each access request to ensure temporary and controlled access.

→ For high-risk access requests, configure an approval process where a supervisor or manager must review and approve the request. This ensures accountability and oversight over who is granted privileged access.

→ Once the task or access duration is complete, the system should automatically revoke permissions. This prevents privileges from persisting longer than necessary, reducing risk.

Step 4: Configure minimal necessary permissions

Objective:
Limit the scope of access even further by applying JEA, ensuring users are granted only the specific permissions necessary for their task without excess. This approach helps manage entitlements effectively, so each role has the precise level of access it needs, reducing the likelihood of privilege escalation.

Process:

→ Identify the minimum access each task requires and configure permissions accordingly.

- For example, instead of recieving full admin rights, a user might only need read access to logs or limited permissions to modify specific configurations.

→ Set up roles specifically designed for certain tasks rather than broad administrative roles. By mapping tasks to roles with limited permissions, JEA ensures access is tailored to job functions.

→ Use contextual factors (such as device type or network location) to further limit access.

- For example, grant write access to certain databases only when they're accessed from the corporate network or VPN.

Step 5: Implement continuous monitoring and real-time auditing

Objective:
Set up real-time monitoring and auditing systems to track all privileged activities, providing visibility and accountability for every action taken with elevated access.

Process:

→ Use security information and event management (SIEM) and PAM solutions to monitor all privileged access requests and actions in real time. These tools can detect unusual access patterns, such as login attempts from unapproved devices or unexpected data transfers.

→ Configure alerts for suspicious behavior.

- For example, if a user attempts to access sensitive files outside normal working hours or from an unusual location, the system should flag this behavior for review.

→ Capture every action taken during privileged sessions, including login times, commands executed, and files accessed. These logs create a traceable record that helps during compliance audits and incident investigations.

→ Conduct periodic audits of access logs to ensure all access requests are justified and appropriate. Look for any patterns that suggest abuse or potential policy adjustments.

💡 **Pro Tip →** When paired with single sign-on (SSO), ZSP can maintain a unified audit trail of access events across all systems accessed through single sign-on. This centralized logging offers a detailed record of each user account's activities, supporting compliance efforts and ensuring that any anomalies are detected promptly.

Common challenges in implementing zero standing privileges

Balancing security with operational efficiency

One of the toughest hurdles with ZSP is finding a balance between stringent access controls and maintaining smooth business operations.

Excessively restrictive access policies or manual processes can slow down workflows. This creates friction, frustration, and even resistance among employees who feel limited in their ability to perform their tasks.

For example, an IT team member needing quick access to resolve an urgent system issue may be delayed by multi-step approval processes, which could lead to system downtime or operational setbacks.

💡 **Solution: Fast-track approval workflows for high-priority requests**

→ For high-priority situations (e.g., urgent troubleshooting tasks), organizations can define triggers that expedite approval workflows.

- For instance, if a request is tied to a critical system error, the system could automatically route the request to a designated approver who can quickly grant temporary access.

→ For frequent tasks, create predefined templates with limited-time access permissions, reducing delays. Once these templates are approved at the policy level, employees can use them without going through repetitive approval processes, as long as the access stays within the template's boundaries.

Gaining buy-in across departments

Employees don't like change — especially if they see it as a burden rather than an improvement.

This gets worse in larger organizations where different departments have unique workflows and may feel ZSP's access restrictions hinder their ability to carry out tasks effectively.

Departments like IT, HR, and finance may have different security needs, leading to concerns that a one-size-fits-all approach will create bottlenecks or limit their ability to operate effectively.

💡 **Solution: Tailored training and communication strategy**

→ Design training programs that address the specific needs and concerns of each department.

- For example, IT staff training could focus on how ZSP helps manage privileged access effectively, while finance could learn how ZSP protects sensitive financial data from unauthorized access.

→ Explain the benefits of ZSP, such as protecting personal and customer data, reducing insider threat risks, and complying with industry regulations. Use practical examples to help employees understand how ZSP safeguards their work without compromising efficiency.

Managing technical complexity

In large organizations, the technical architecture can be complex, often involving a mix of legacy systems, hybrid environments, and multiple applications across different platforms. Implementing ZSP in this type of environment can be difficult.

For example, legacy systems may lack native support for ZSP and require specialized integration. Another technical challenge is maintaining uniform ZSP policies across cloud and on-premises environments, which often operate under different configurations and security protocols.

As a result, the more complex the infrastructure, the higher the risk of access gaps, policy inconsistencies, or compatibility issues—all of which can undermine the effectiveness of ZSP.

💡 **Solution**: **Phased rollout and integration with existing security infrastructure**

→ Begin by implementing ZSP in areas that present the highest risk, such as IT administrators, database managers, or cloud services handling sensitive data. Gradually extend ZSP policies to other departments or systems to minimize operational disruptions.

→ If a legacy system doesn't support ZSP directly, establish compensating controls, such as network segmentation, endpoint monitoring, and strict logging. These measures help contain security risks and provide oversight, even if the system can't fully support ZSP.

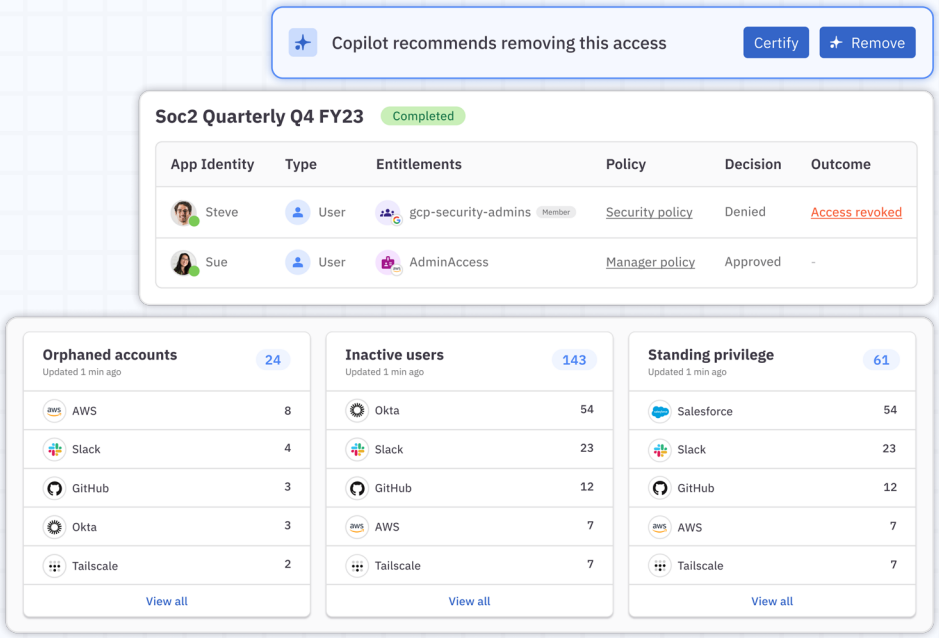
Make zero standing privileges work for your organization

Every year, businesses report an increase in identity-based cyberattacks on their digital infrastructure.

From compromised admin credentials to insider threats, these breaches share a common thread: excessive standing privileges.

While traditional “always-on” access privileges might have sufficed in the past, they’re simply a liability for businesses now.

This is where ConductorOne shines as a comprehensive solution for privilege management.



Instead of wrestling with manual processes and lengthy approval chains, imagine your team automatically accessing resources exactly when they need them, for exactly as long as they need them.

ConductorOne’s streamlined workflows mean your developers spend more time coding and less time waiting for permissions. Your security team gains comprehensive visibility without becoming a bottleneck, while automated access reviews ensure compliance without the quarterly scramble.

For organizations juggling a complex mix of applications, ConductorOne’s centralized access management simplifies the entire process.

With visibility across all your systems, you can confidently control access levels, monitor usage, and automate approvals—all while maintaining a consistent ZSP framework.

🔗 How Ramp Implemented Least Privilege Access: **Read Case Study →**.

Ready to learn more?

Talk to our team to learn how ConductorOne can help you implement ZSP.