sonraí security

SONRAI SECURITY | WHITEPAPER

# Least Privilege Without the Ticket Queue

How the Sonrai Cloud Permissions Firewall enforces least privilege at the organization level - without cleaning up entitlements one identity at a time

The standard answer to cloud over-permissioning has been CIEM: inventory every identity, compute its effective and used permissions, score the findings, and open tickets asking developers to rewrite IAM policies. The model is analytically sound and operationally broken. Roughly 90% of identities are over-privileged. Ticket-driven remediation means thousands of bespoke policy edits, each carrying breakage risk and competing with feature work. Most will be undone by drift the following quarter. Most organizations that bought CIEM are no closer to least privilege than when they started.

The Sonrai Cloud Permissions Firewall (CPF) takes a different enforcement point. Instead of fixing identity policies one at a time, it deploys preventative deny controls within the AWS organization hierarchy - Service Control Policies (SCPs) and Resource Control Policies (RCPs) at the scope you choose - governing only the permissions that matter, with every legitimate existing user of those permissions automatically exempted based on historical usage. Least privilege stops being a backlog and becomes a default posture: 100% of identities protected by default, on day one, with no disruption to running workloads.

## Why the CIEM Ticket Model Stalled

CIEM tools are good at diagnosis. They will tell you, accurately, that a role has 2,900 permissions and uses 40. The failure is in the prescription: a finding becomes a ticket, the ticket goes to the team that owns the identity, and that team is asked to hand-author a tighter IAM policy. At cloud scale, this fails for predictable reasons.

First, the volume is unworkable. When 90% of identities are over-privileged, remediating each one requires someone who understands both IAM policy language and what that identity and its workloads actually do. Second, every edit is risky: which permissions a workload depends on isn't obvious; thus, to avoid breaking something in production, teams defer and tickets age. Third, the target moves. New identities, new services, and new deployments regenerate the finding set faster than teams can close it. The result is a permanent remediation backlog that measures activity in tickets closed rather than risk removed. The environment is never actually protected, and the backlog is permanent.

## Enforce at the Organization, Not the Identity

The Cloud Permissions Firewall inverts the model. Rather than shrinking each identity's policy to match its usage, it deploys deny controls at the organization level - at the Org, an OU, or account - that block privileged permissions for everyone except the identities that demonstrably need them. Exemptions are expressed as conditions in the deployed policy (keyed on aws:PrincipalTag and aws:PrincipalArn), not as one statement per identity. When access needs change, an approval flips an attribute such as a tag on the identity, which satisfies a condition already present in the deployed SCP — the policy document itself is never rewritten.

Control stays entirely inside your account. Sonrai generates IaC (CloudFormation, Terraform, etc...) containing the SCPs and RCPs, and your team deploys it; Sonrai has no direct access to your organization's policies. Enforcement is preventative rather than advisory. The outcome is a deployed deny policy, not a ticket - and it is reversible at any time.

What follows are the properties that make this practical to run in production, and that the ticket-driven model could never deliver.

> **Least privilege stops being a backlog and becomes a default posture: 100% of identities protected by default, on day one, with no disruption to running workloads.**

### No individual cleanup of existing entitlements

CPF never asks you to rewrite the IAM policies you already have. Existing roles, users, and their attached policies stay exactly as they are — the organization-level deny simply sits above them and neutralizes the unused privileged permissions they contain. One control deployed at an OU does the work of hundreds of individual policy edits, and there is no migration project, no policy refactoring sprint, and no dependency on developer time to reach a protected state.

### Good-enough least privilege: control what is actually dangerous

Perfect least privilege - every identity trimmed to exactly its used actions- is the standard CIEM chased and never reached. CPF targets what matters instead: Sonrai maintains a catalog of privileged permissions, the actions that enable privilege escalation, lateral movement, and exposure, and the default controls govern only those. Low-risk actions remain allowed, which means an engineer browsing the AWS console does not hit a wall of denials; list and describe operations keep working, and day-to-day operations are untouched. You get the risk reduction of least privilege without the operational friction that causes least-privilege programs to be rolled back. Custom controls can extend the catalog where your environment needs it (like preventing data-destruction actions and other company-specified initiatives).

### Zero disruption by design: historical analytics drive default exemptions

Before anything is enforced, CPF builds a complete map of every identity - users, roles, workloads, and agents - to the permissions and resources it actually uses. When a control is

deployed, every identity that has performed a governed action within the previous 90 days is automatically exempted. The practical consequence: deploying a control against an estate where 90% of identities are overprivileged disrupts nothing, because the identities actively using a permission keep it and only the unused grants are cut off. This is what makes it safe to run the controls as-is on day one, rather than after months of analysis and sign-off.

## Coverage for AI agents, with a human back in the loop

AI agents now hold cloud identities of their own, acting directly or on behalf of humans, and they inherit the same over-permissioning humans do, albeit with less predictable behavior. CPF governs agent identities exactly like any other principal: an agent gets the privileged permissions its history justifies and nothing more. When an agent steps outside that envelope - because it has gone off-task, because its credentials have been hijacked, or because you are facing a red team or attacker operating a frontier-class agent - the attempted action is denied, and an approval request is raised automatically. EventBridge rules detect workload- and agent-triggered denials and route them to the right approvers for that part of the cloud immediately, putting a human decision between an autonomous system and a privileged action at precisely the moment it matters.

> **You get the risk reduction of least privilege without the operational friction that causes least-privilege programs to be rolled back.**

## Pattern-based exemptions: golden pipelines are never interrupted

Deployment pipelines need to wield privileged permissions, and no security control that breaks the release train survives contact with engineering. CPF supports exemptions based on patterns - matching your pipeline roles by principal pattern rather than enumerating them - so golden pipelines are never interrupted as new pipeline instances come and go. Alternatively, for tighter governance, you can treat pipeline elevation as a just-in-time access request: the pipeline requests it, an owner approves, and the grant is time-bound.

## Safe quarantine of zombie identities

Zombie identities - dormant, unused roles and users left over from past projects - expand the standing attack surface while providing no operational value. CPF quarantines them with an organization-level deny rather than deleting them: the identity is instantly unusable to an attacker, but nothing is destroyed. If a quarantined identity turns out to be needed, it is restored with an approval rather than an infrastructure archaeology project. Quarantine gives you the risk reduction of deletion with none of its irreversibility.

## Permissions on Demand: when things change, approval is easy

Least privilege fails when getting new access is painful. Teams respond by hoarding permissions. With CPF, when an identity needs a permission it does not have, the request arrives in Slack or Microsoft Teams (or email) and the approver actions it in place. Grants can be time-limited, self-approval can be permitted for designated low-risk scopes, and you can designate certain permissions to always require a second party. On approval, Sonrai updates an attribute such as a tag and the existing policy condition takes effect immediately - no policy change, no deployment, no ticket. Unanswered requests escalate to the approver at the next

level up the scope hierarchy, and a request that reaches the end of the chain unapproved is simply denied: the system fails closed.

### Everything has an owner

Every scope in the firewall - root, OU, account - has a designated owner and approvers, established when the firewall is set up. That single fact fixes the accountability gap that ticket-driven remediation never solved: when a permission change is requested, the person who owns that part of the estate sees it, decides it, and the decision is recorded. Teams always know when permissions in their scope are changing and who approved it, and access decisions are made by the people with context instead of a central security queue.

## From Zero to Enforced: The Five Steps

1. **Discover.** A security-auditor-style role builds a map of every identity — users, roles, agents, and workloads — to every permission and resource. That map defines the allow list, which is what prevents breaking existing workloads and operations.
2. **Delegate.** Owners and approvers are established per scope, along with the conditions governing access requests: where self-approval is allowed, time limits on grants, which permissions always require a second party, and how unanswered requests escalate.
3. **Decide.** Pick a scope — root, OU, or account — and a control. The console shows the identity blast radius before anything is deployed.
4. **Deploy.** Sonrai generates CloudFormation containing the SCPs and RCPs, and your team deploys it. Sonrai has no direct access to the policies in your organization.
5. **Operate.** Requests flow through Slack or Teams. Workload- and agent-triggered requests are auto-detected by EventBridge rules and notify approvers immediately. On approval, Sonrai changes an attribute such as a tag, which satisfies a condition already in the deployed SCP — the policy document is never rewritten.

## Conclusion

The CIEM generation proved that visibility alone does not change risk: a finding without an enforcement mechanism is a ticket, and tickets lose to feature work indefinitely. The Cloud Permissions Firewall makes least privilege an enforced default rather than an aspiration. It eliminates the work of individual identity cleanup, broken consoles, disrupted workloads, and interrupted pipelines. It places a human approval in front of every privileged escalation, whether it comes from an engineer, a workload, or an AI agent. Organizations deploy it and reach a protected state in days, with 100% of identities protected by default. The neverending backlog is eliminated. Cloud teams stop chasing a definition of least privilege ceaselessly out of date, and instead start managing a future-proof default deny state.