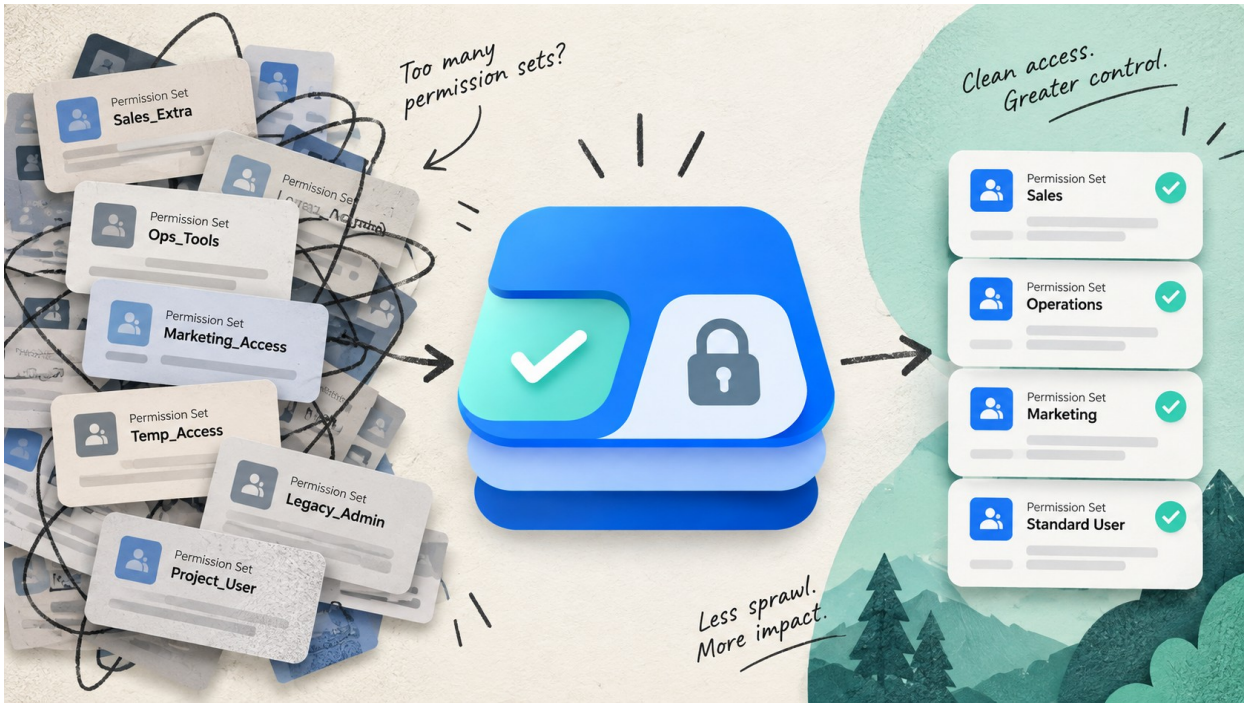


Permission Set Sprawl: How to Clean Up Without Breaking Access

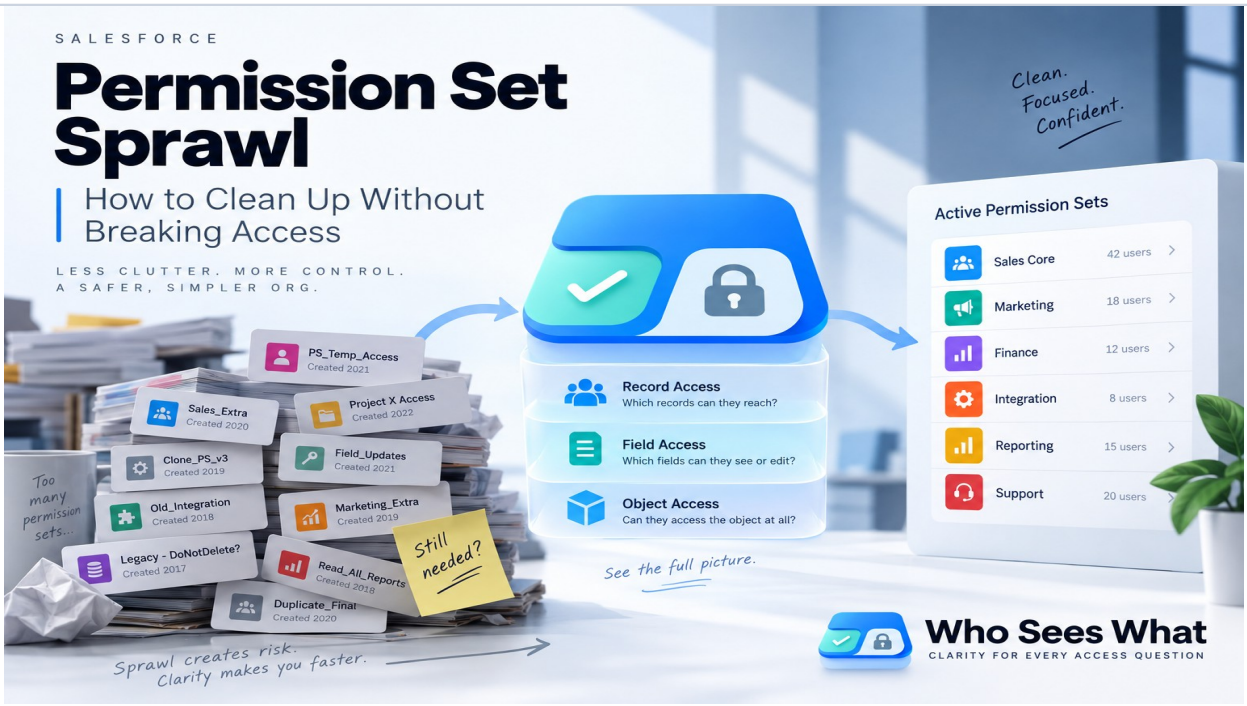
A permission set that looks redundant cannot be safely deleted based on its name.

> Short answer: a permission set that looks redundant or overlapping cannot be safely deleted based on its name or description. It has to be checked against who is currently assigned and what those specific users would lose, because permission sets accumulate faster than they get retired.

How does permission set sprawl happen?

Sprawl accumulates the same way over-permissioning does: through individually reasonable decisions with no natural review moment. A permission set gets created for a temporary need and never deleted once the need ends. A permission set gets duplicated with minor variations rather than consolidated, because copying is faster than building something flexible. Permission set groups get built on top of an already sprawling base rather than replacing it.

Three years into this pattern, a mature org commonly has dozens of permission sets with overlapping or unclear purposes, named by people who have since left.



Why is sprawl not merely untidy?

Every additional permission set is another place a grant can exist that nobody remembers checking. When investigating an access question, a longer list means a longer investigation. When running a review, more permission sets mean more surface area to confirm is still correct.

Why is deleting based on the name unsafe?

A permission set named "Legacy - Old Sales Process" sounds safe to delete. Whether it actually is depends entirely on whether anyone is still assigned to it and what they would lose, which the name does not tell you. Permission sets frequently outlive the accuracy of their own names.

What does a safer cleanup process look like?

Step 1: inventory what exists

List every permission set and group, with last-modified date and, if available, last assignment date.

Step 2: for each candidate, find who is currently assigned

If the list is empty, it is safe to remove immediately. If not, every assigned user needs individual consideration.

Step 3: for each assigned user, confirm what they would lose

This determines whether a permission set is truly redundant or only appears that way. If a user’s other permission sets already grant everything the candidate grants, removal changes nothing for them.

Step 4: consolidate before deleting, where multiple permission sets overlap

Build a single permission set covering the union of what is actually needed, migrate assigned users, then retire the originals, rather than picking one to keep and accidentally losing access some users genuinely need.

Step 5: remove in stages, and verify after each stage

This limits the blast radius of a mistake to one stage rather than the whole cleanup.

What makes step 3 practical at any real scale?

Step 3 has a real cost per user under manual investigation. A permission set assigned to forty users means forty individual checks. Who Sees What makes this tractable by resolving any user's full effective access in seconds, including exactly which permission sets are contributing which grants.

Get Who Sees What free on the AppExchange →

Who Sees What is built and maintained by TwinStack Solutions, a Salesforce partner. Questions about setup: info@twinstack.net