7 Moments When You Need to Know a User's Effective Access

The same underlying question, asked in seven different contexts.

> Short answer: a user reports missing access, a user has access they should not, a scheduled review, onboarding, offboarding, permission set cleanup, and a consulting handover all reduce to the same question: for this person, what can they access, and what specifically grants it.

1. A user reports they cannot see something they expect to

The most common trigger. Answering it means checking object access, field-level security, and record-level sharing, since any of the three could be the cause and the symptom looks identical regardless.

2. A user has access they should not have

The less visible counterpart. Nobody files a ticket to report they can see something. This surfaces through incidents or spot checks, and confirming it requires the same investigation as any other access question, just triggered by suspicion rather than complaint.

3. A scheduled or compliance-driven security review

The value of a review is proportional to how completely it covers the user base, and completeness is exactly what a thirty-minute-per-user cost makes impractical at scale.

4. Onboarding a new hire

Confirming assigned permission sets actually produce the intended access, before the first day rather than after the first complaint.

5. Offboarding a departing employee

The highest-stakes verification, since the failure mode is a departed employee retaining access with nobody left who has a reason to notice.

6. Cleaning up permission set sprawl

Before deleting a permission set that looks redundant, confirming who is assigned and what they would lose without it.

7. Documenting an org's access model during a handover

Being able to look up any user and show exactly what they can reach gives an incoming admin a concrete, checkable starting point.

What is the pattern underneath all seven?

Every one of these is the identical question, asked in a different context. None starts from a permission set or a profile, because none of the seven starts from a permission set or a profile. They all start from a person.

That is the entire premise behind Who Sees What: resolve the question in the direction it is actually asked. Pick the person, see everything they can reach across object, field and record level, and see the specific rule responsible for each grant.

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