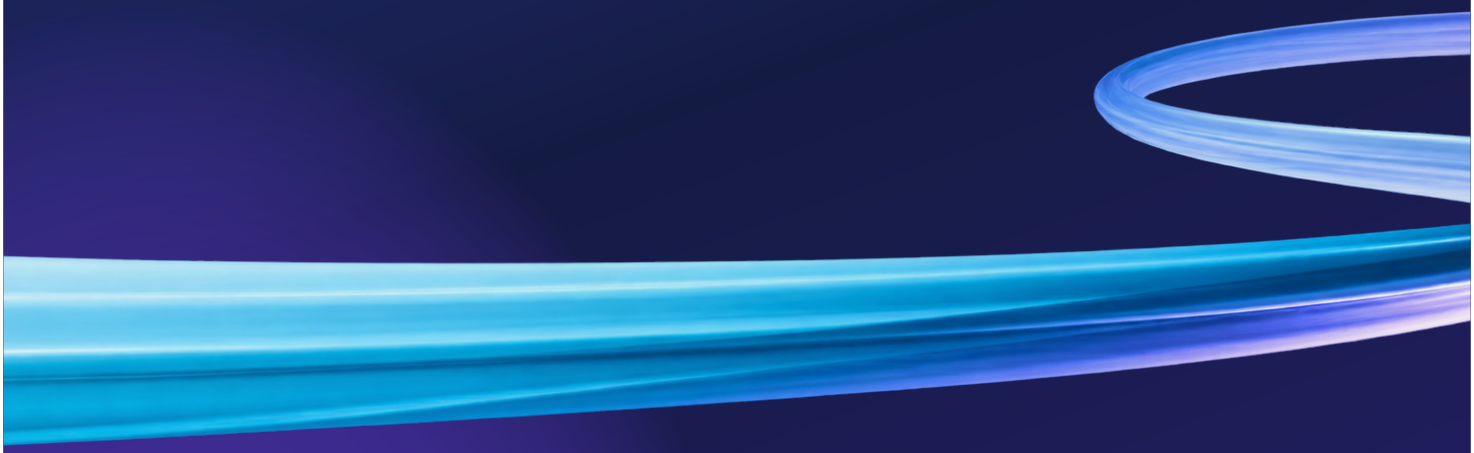
xtype | servicenow

SERVICENOW ACCESS GOVERNANCE · AN XTYPE E-BOOK

Borrowed Power

Finding access abuse in your ServiceNow, and proving what you find



WHAT'S INSIDE

Contents

	Introduction: Four shapes, and the proof	3
1	Nobody breaks in. They log in.	4
2	The four shapes of borrowed power	6
3	The honest page: volume, and the problem of “normal”	11
4	From suspicion to proof	14
5	Every flag gets an answer	17
	Checklist: Detection and evidence	18
	Start this week	19
	About xtype	20

Product screens throughout are from xtype Compliance for ServiceNow. Where a screen is marked as sample data, it shows the product's real views populated with illustrative records.

INTRODUCTION

Four shapes, and the proof

The access abuse that hurts your organization will not come from a hacker breaking in. It will come from a real, approved account doing something it should not do. This kind of abuse always takes one of **four recognizable shapes of borrowed power**.

But spotting the shape is not enough. In a real ServiceNow environment, thousands of small actions happen every day. Unless you know what "normal" looks like, and keep a record that turns suspicion into proof, the warning signs get lost in the noise.

This e-book covers both halves of the problem: **how to recognize borrowed power, and how to prove what you found.**

IN THIS E-BOOK

Part 1 shows why the accounts that hurt you pass every login check.

Part 2 walks through the four shapes: the self-granted role, the temporary grant, the impersonation and the widened rule.

Part 3 deals with volume, and with learning what "normal" looks like.

Parts 4 and 5 turn a flag into evidence, and evidence into a recorded decision.

01 PART ONE

Nobody breaks in. They log in.

Picture the security event that will hurt your organization. It will not set off an intrusion alarm. There will be no malware and no warning banner. It will be a real account that your team created, with permissions someone approved, but doing something it should not do.

This is not a guess. It is how real attacks happen today. Attackers call the help desk and talk someone into resetting a password. They buy working login details instead of trying to break encryption. And people inside your own company do not need to "attack" anything at all. They take a shortcut, give themselves more access than they need, and plan to undo it before anyone notices.

In every case, the account passes every login check the platform has, because the account is genuine.

Only the behavior is wrong.

Now add a second problem, one that often arrives months later. Someone, such as an auditor, a regulator or your own security team, asks a simple question: who gave this person access, when, and why?

For most ServiceNow environments, the honest answer is a manual data pull, a spreadsheet, and the hope that the right information turns up. ServiceNow instances are kept separate from each other by design, so this work multiplies as data is pulled from development, QA and production to trace the activity. And some of ServiceNow's built-in change history is deleted on its own schedule, sometimes within days. Often, the exact record you need no longer exists.

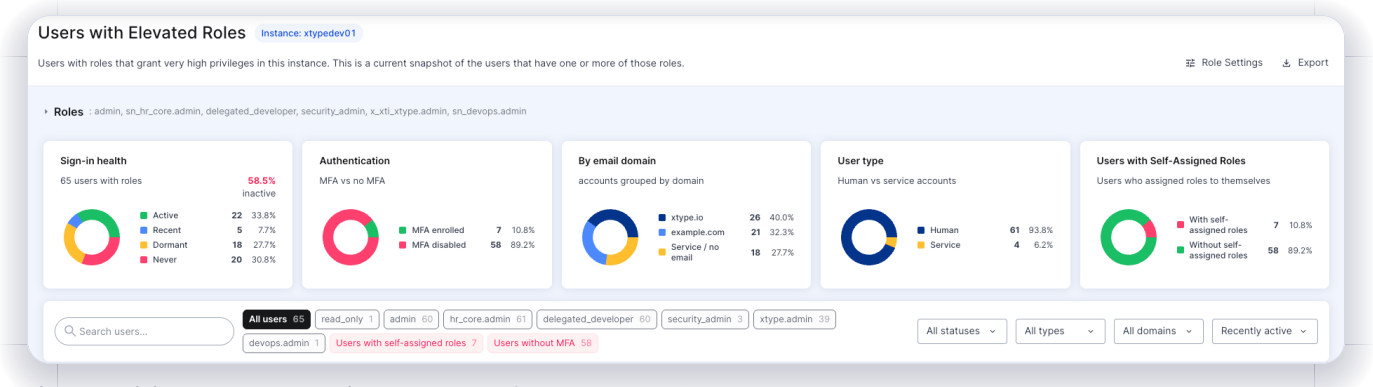


Figure 1: Privileged access overview. A live view of one ServiceNow instance in the Users with Elevated Roles dashboard: 65 accounts hold elevated roles, and more than half of them are dormant or have never signed in. This is what an environment looks like before anyone misuses anything: a lot of unused power, mostly unwatched.

Both problems share one root cause: on the platform that runs your most critical work, access can change quietly and **leave no record behind.**

The good news: borrowed power is not random. It uses the platform's own features, so it always takes one of a small number of shapes. Four, in practice.

Learn these four shapes. Make them visible. Keep a record that proves what you found.

That is the whole approach, and the rest of this e-book walks through it.

02 PART TWO

The four shapes of borrowed power

1 SHAPE 1 OF 4

The self-granted role

Someone gives a powerful role to themselves. Often this is innocent, such as an administrator fixing a problem at midnight, who grants access first and plans to ask permission later. But **no compliance framework** accepts a person approving their own access increase. And an attacker who takes over an admin account will typically do exactly this first.

The real question is never "is self-granting always bad?" It is: **"did anyone notice, and did anyone ask why?"**

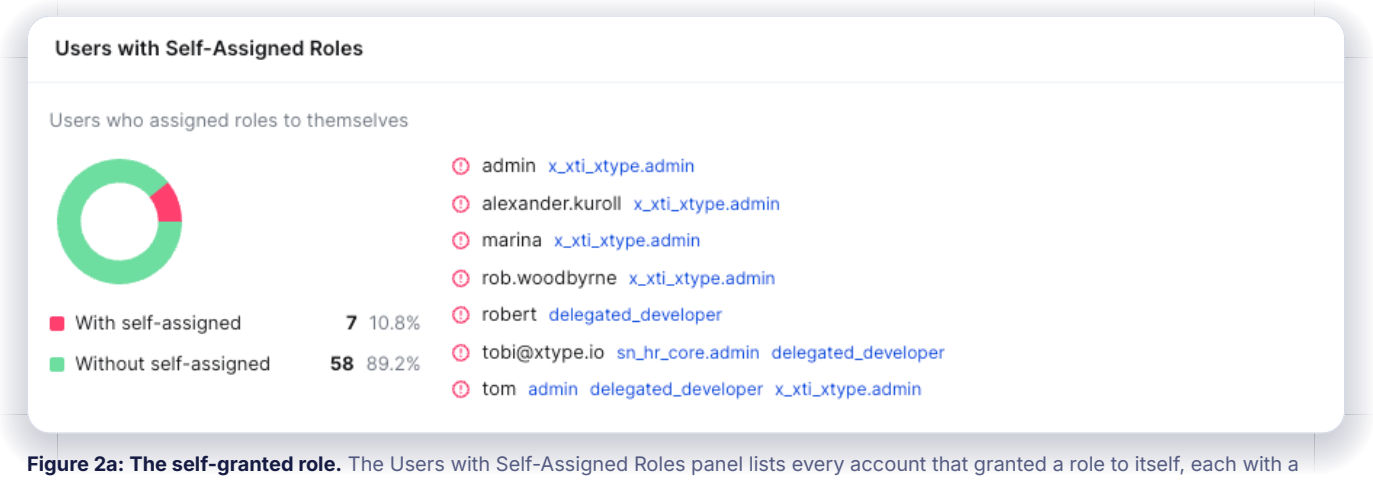


Figure 2a: The self-granted role. The Users with Self-Assigned Roles panel lists every account that granted a role to itself, each with a warning marker. Note that admin, the platform's own administrator account, appears in its own list.

2 SHAPE 2 OF 4

The temporary grant

A role is granted on Friday evening, used over the weekend, then removed on Sunday night. By Monday's review, the access list looks completely normal: the power was borrowed and returned before anyone checked.

Of the four shapes, this one is the hardest to catch by hand. The evidence is not something that still exists; it is something that existed only briefly. This pattern is often a sign that someone knew they were working around a control.

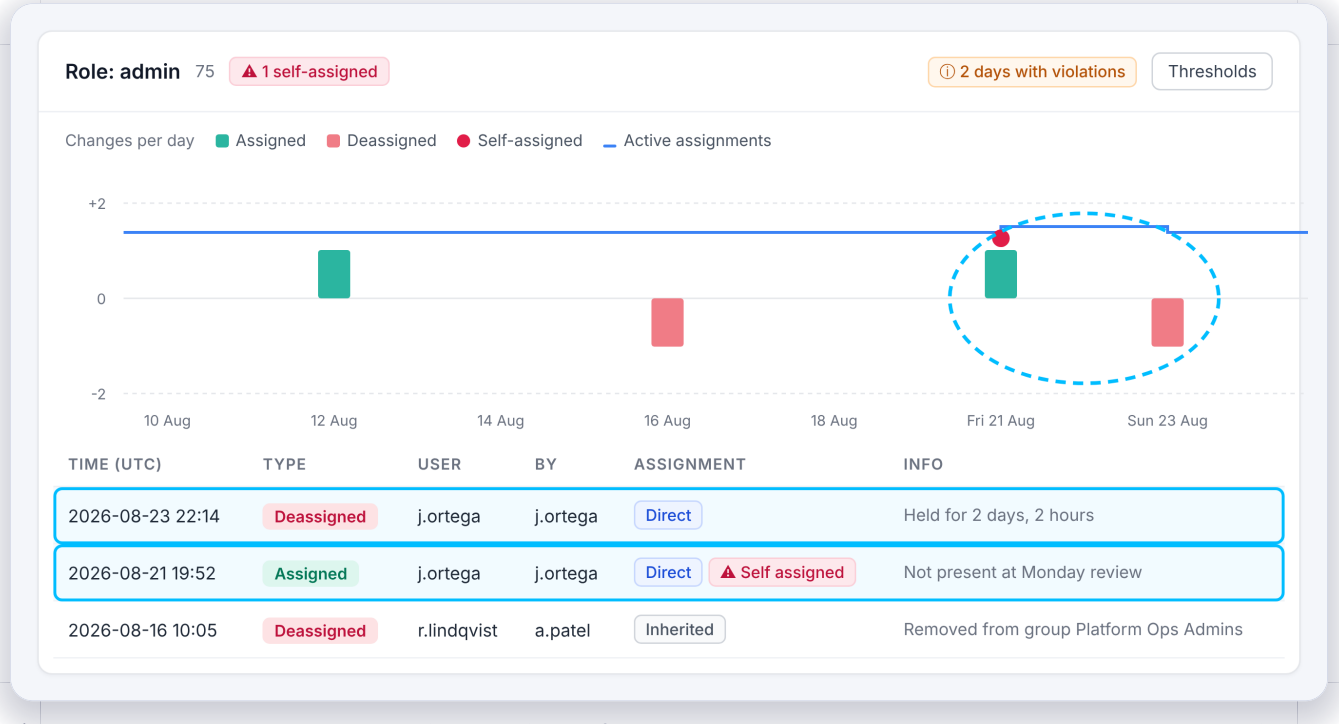


Figure 2b: The temporary grant. A Role Assignments timeline for admin: the same user grants the role to themselves on Friday evening and removes it on Sunday night. Both events stay on the record, with timestamps, even though Monday's access list looks clean. *Sample data for illustration.*

3 SHAPE 3 OF 4

The impersonation

An administrator can temporarily act as another user. This is a legitimate, built-in support feature. But it is also a privilege event wearing someone else's name. While impersonating, everything the administrator does looks, at first glance, as though the other person did it.

Most organizations review who holds admin rights. Far fewer review who those admin rights have pretended to be.

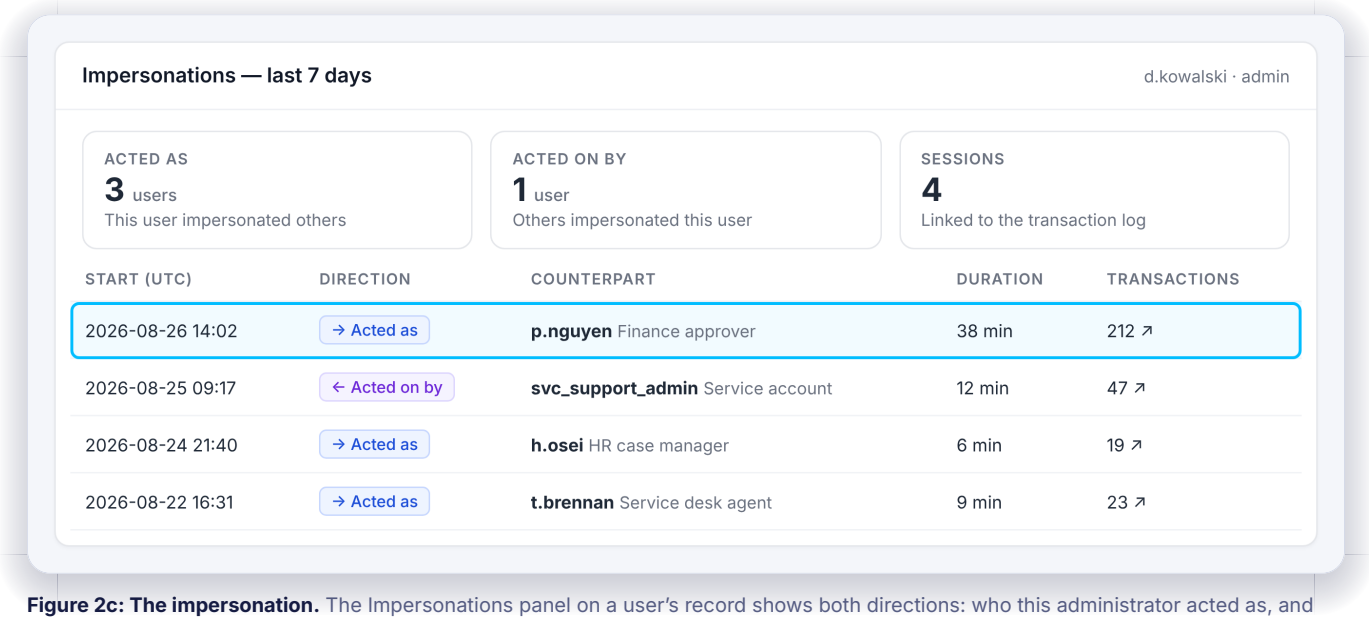


Figure 2c: The impersonation. The Impersonations panel on a user's record shows both directions: who this administrator acted as, and who acted as them, with session length and a link to the transactions performed. Impersonation history is available on ServiceNow instances running the Australia release or later. *Sample data for illustration.*

4 SHAPE 4 OF 4

The widened rule

This is the hardest shape to notice. Beneath the roles everyone talks about, ServiceNow runs on tens of thousands of small access rules (ACLs), each one a gatekeeper deciding whether a given role may read, create, change or delete a specific kind of record, sometimes down to a single field. Roles say *who you are*; these rules say *what that gets you*. Change a rule, and you've changed the meaning of every role it touches.

WHY ACL CHANGES ARE USUALLY LEGITIMATE

These rules change constantly, and almost always for good reasons. Every new application ships with its own set. Platform upgrades revise hundreds of them at a time. Developers widen a rule so a new integration can write to a table, or so a service portal can display a field it couldn't before. This is ordinary work when done by the right people.

That legitimacy is exactly what makes this shape subtle: a widened rule doesn't look like an attack, because 99% of the time it isn't one. The problem is the 1%: a rule quietly loosened outside any release, by someone whose reasons no one asked, hiding in a category of change that everyone has learned to wave through.

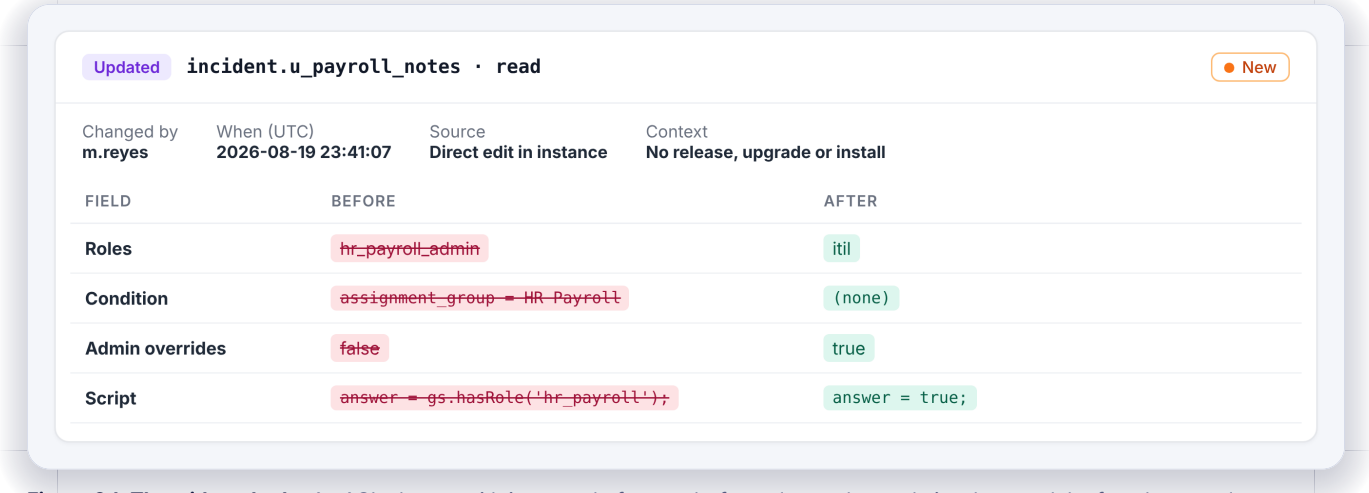


Figure 2d: The widened rule. An ACL change with its exact before-and-after values: who made it, when, and the fact that no release, upgrade or install explains it. *Sample data for illustration.*

Every review that only checks individual roles will pass. Nothing on any single user's record moved.

But the actual power still grew.

> **Every real security incident we have heard about fits one of these four shapes. The shapes themselves are not hard to understand. The hard part is spotting them inside everything else that is happening.**

WHY AUDITORS CARE

What the frameworks require

SOX

SECTION 404 · ITGC

Access to financially relevant systems must be authorized, segregated by duty and monitored. A self-granted role or a widened rule is exactly the kind of unsegregated access this control is designed to catch.

SOC 2

COMMON CRITERIA CC6.3

Access must be authorized, modified and removed based on role and least privilege, "giving consideration to" segregation of duties. Self-granting bypasses this control by definition.

ISO 27001:2022

ANNEX A 8.2

Privileged access rights must be restricted, formally authorized and reviewed on a regular cadence: the control most often cited for temporary grants and impersonation events that never appear in a scheduled review.

03 PART THREE

The honest page: volume, and the problem of “normal”

Here is the part most vendors do not tell you: knowing the four shapes does not make them easy to find. Anyone who says a tool makes this effortless is overselling.

Look at the real numbers from a single, ordinary ServiceNow instance:

70,000
access rules

20,000
actions in a week from one busy administrator

500
legitimate access-rule changes in a single hour of a routine platform upgrade

Inside that one hour of upgrade activity, a dozen malicious changes would be invisible to any human reviewer. And the temporary grant, the weekend borrow, is almost invisible by definition: by the time you look for it, it is already gone.

Volume is not just an inconvenience here. Volume is what hides the problem.

The right tools cannot remove this problem, but they can make it manageable, in two specific ways.

First: the system should label its own routine activity

When the platform knows that an upgrade or a plugin installation caused a burst of changes, it should say so. It can tag the burst, explain it, and let a reviewer approve it in seconds. Then a burst of changes with no label becomes the real finding, not because it is large, but because nothing explains it.

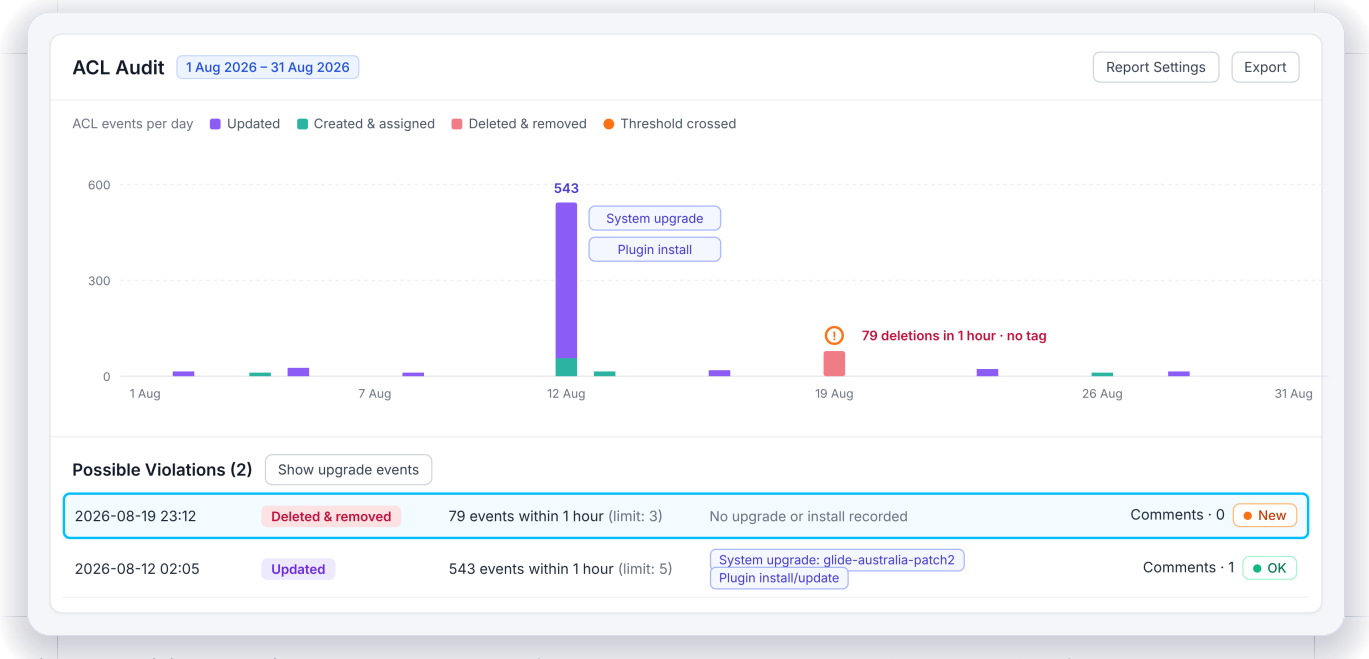


Figure 3: Activity chart with context tags. A month of access-rule changes in the ACL Audit report. A spike of 543 changes carries clear tags: system upgrade, plugin install. Routine. Beside it, a separate entry shows 79 deletions in one hour against a threshold of three, with no tag at all. No upgrade. No install. That is the finding worth a closer look. *Sample data for illustration.*

Second: “normal” has to be learned, not guessed

How many new grants to a powerful role is normal for your organization? How fast is too fast? There is no single answer that works for every company.

The practical method: look at your own last three months of activity, and set simple threshold rules just above that level:

> This many new role holders within this time window
>
> This many removals
>
> Any grant reversed within a day

These thresholds are not artificial intelligence. They are simply **your own history, turned into a tripwire.** And this only works if something has actually been keeping that history.



Normal is not a setting you configure once. It is a fact you learn, and only if you are already keeping the record that shows it.

04 PART FOUR

From suspicion to proof

A flag without a record is just a guess. The moment something suspicious appears, such as a self-grant or an unlabeled burst of changes, the questions begin. They are always the same five:

1. **Who did it?**
2. **What exactly changed?**
3. **When did it happen?**
4. **How did they have the power to do it?**
5. **Can you answer all of this for the entire time period in question, not just for today?**

This is where the two halves of this e-book meet, and where the right tools matter most, so this is where we introduce ours. The screens shown throughout this e-book are from **xtype Compliance for ServiceNow.** The reason they can answer the five questions above is not clever analysis. It is something simpler, and harder: **xtype keeps the record.**

A record that can stand up to an audit needs a specific structure

- **Attributable events.** Every change is tied to a person (who, what, when, and under whose authority), not just a comparison between two exports taken at different times.
- **Grant lineage.** Every access grant shows its origin: given directly, or inherited through a group, and in either case, from whom.
- **The ability to look back in time.** "What did this look like in March?" should be a simple search, not a research project.
- **A stated start date.** A record that states exactly when its coverage began is credible evidence. A claim of "unlimited history" is not.



Figure 4: Grant lineage and history. One user's record answering the auditor's real question, not "who has admin" but "how did they get it": a direct grant by the system in 2022, then an inherited, self-assigned grant on August 25. Beside it, three months of grants and removals for the same role. *Sample data for illustration.*

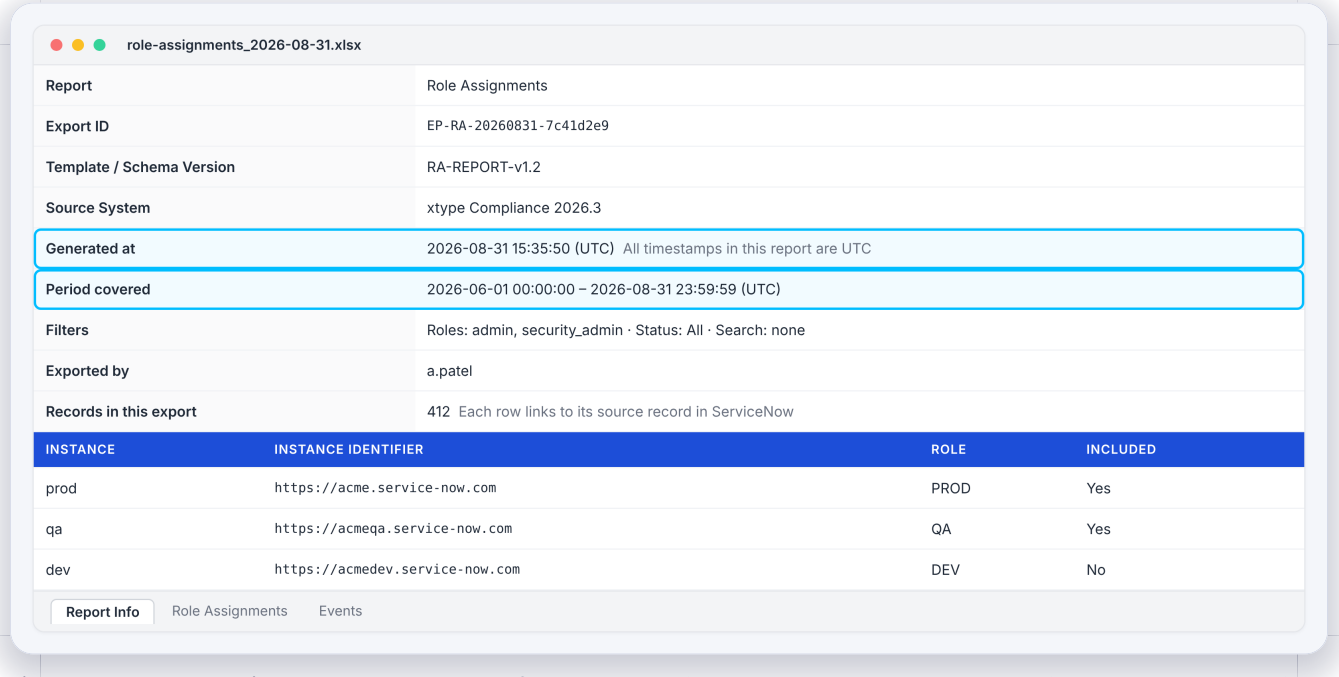
When a report leaves your system, the file must carry its own proof

An exported report should record:

- A unique export ID
- When it was generated, and in which time zone
- What time period it covers (a separate detail from the generation date)
- Which ServiceNow instance it came from, and which filters were applied
- Who exported it, and how many records it holds, with every row linked back to its source

This is the difference between a screenshot and real evidence:

A screenshot asks to be trusted; evidence explains itself.



role-assignments_2026-08-31.xlsx

Report	Role Assignments
Export ID	EP-RA-20260831-7c41d2e9
Template / Schema Version	RA-REPORT-v1.2
Source System	xtype Compliance 2026.3
Generated at	2026-08-31 15:35:50 (UTC) All timestamps in this report are UTC
Period covered	2026-06-01 00:00:00 – 2026-08-31 23:59:59 (UTC)
Filters	Roles: admin, security_admin · Status: All · Search: none
Exported by	a.patel
Records in this export	412 Each row links to its source record in ServiceNow

INSTANCE	INSTANCE IDENTIFIER	ROLE	INCLUDED
prod	https://acme.service-now.com	PROD	Yes
qa	https://acmeqa.service-now.com	QA	Yes
dev	https://acmedev.service-now.com	DEV	No

Report Info | Role Assignments | Events

Figure 5: The export's evidence sheet. The Report Info sheet included in every export: export ID, generation timestamp versus the period covered, population, filters and exporting user. A file an auditor can question and check. *Sample data for illustration.*

05 PART FIVE

Every flag gets an answer

Detection and evidence still leave one gap open: the human decision. A queue of findings that nobody reviews is just a source of worry with a nice interface.

The final step is simple to describe, and powerful in practice: **every flag needs an answer, with a status and a written reason.**

That includes an honest “OK”

Some flagged events are real, and still acceptable: for example, a documented emergency change, or an approved emergency-access grant with its paperwork on file. A review system that only allows “violation” or “false alarm” as answers forces reviewers to choose the wrong label.

Recording “this happened, and here is why it was acceptable” is not lenience. It is exactly the reasoning an auditor wants to see.

The goal was never zero flags. The goal is zero flags without an explanation.

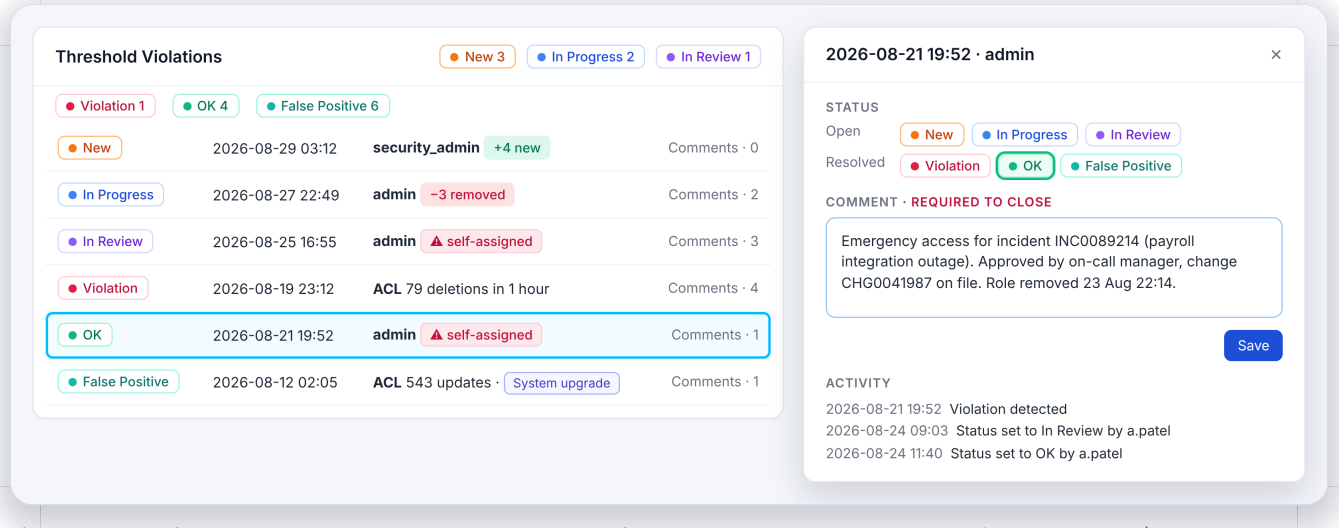


Figure 6: The review queue. Flagged events with clear statuses (new, in progress, in review, violation, false positive, OK), each requiring a written comment before it can close. The reviewer’s reasoning becomes part of the permanent record. *Sample data for illustration.*

CHECKLIST · DETECTION · EVIDENCE · REVIEW

The Borrowed-Power Detection and Evidence Checklist

Put detection, evidence and review together, and borrowing power stops being safe: not because it becomes impossible, but because it becomes visible, easy to investigate and permanently recorded. Use this page to test your own ServiceNow stack.

A The four shapes, and the question to ask about each

- [ ] **Self-granted role.** Who elevated themselves in the last 90 days, and where is the explanation for each?
- [ ] **Temporary grant.** Which powerful roles were granted and removed within a short window, and what happened in between?
- [ ] **Impersonation.** Who has acted as your privileged users, and who have your privileged users acted as?
- [ ] **Widened rule.** Which access rules changed, and for updates, what were the exact before-and-after values?

B Starter thresholds

TUNE AGAINST THREE MONTHS OF YOUR OWN HISTORY

>1
new holder of a monitored role within 5 minutes
FLAG

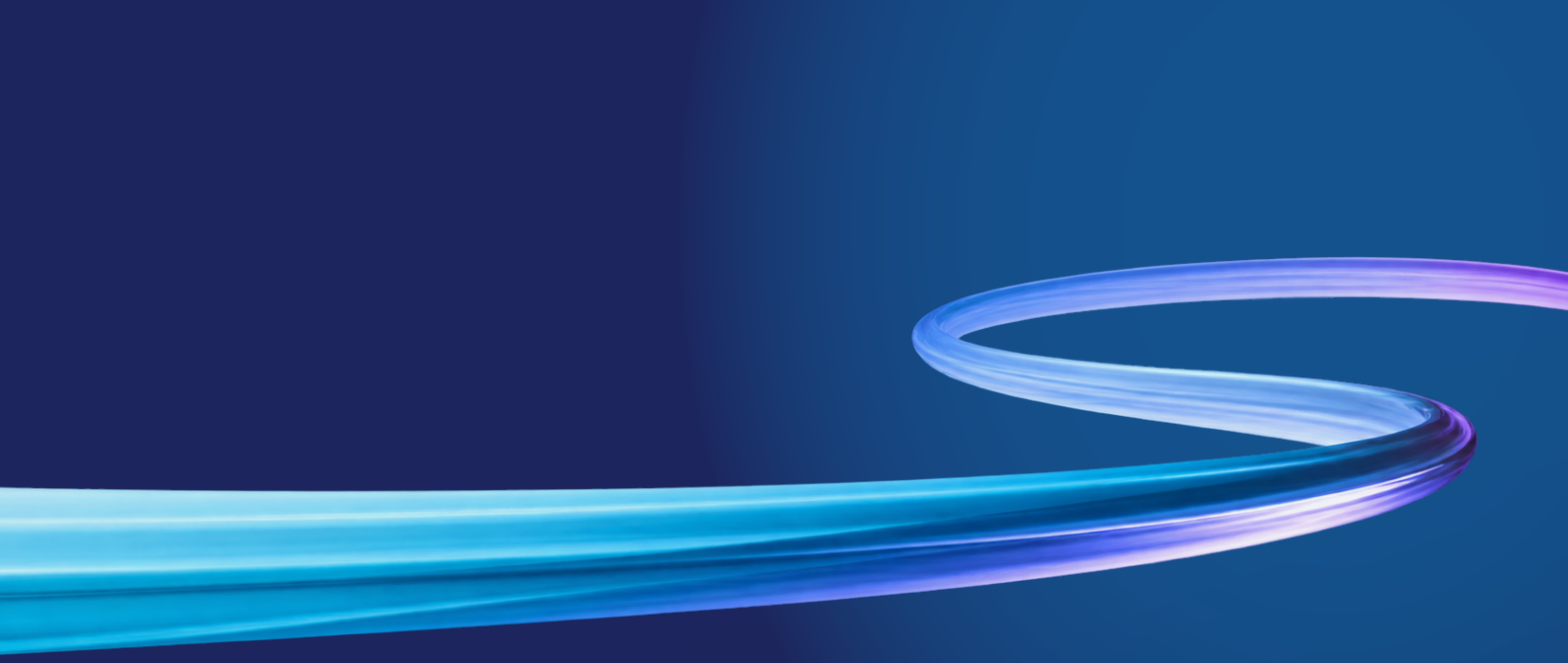
>1
removal within 5 minutes
FLAG

1 day
any grant reversed within this window (the temporary-grant pattern)
FLAG

C Five things a defensible access record must contain

- [ ] **Attributable events.** Who, what, when and by whom, recorded per change rather than per snapshot.
- [ ] **Grant lineage.** Direct or inherited, and through which group.
- [ ] **Both clocks.** When it happened, and when it was found.
- [ ] **A stated coverage start date.** Honesty about the record's own edges.
- [ ] **Reviewed decisions with written reasons.** Including the honest "OK."

Detection plus evidence is what creates real deterrence.



START THIS WEEK

Pick your most powerful role in ServiceNow

Ask the four-shape questions about the last ninety days. If you cannot answer them in a single afternoon, that is not a personal failure. It means you are missing a record.

Start keeping one.

See xtype Compliance in action →

About xtype

xtype is a governance platform built for ServiceNow. It gives enterprises a single, real-time view of every change, access decision and configuration, across every instance they run, so that governance is provable, not reconstructed after the fact.

xtype Compliance, the product behind the reporting shown in this e-book, turns that record into audit-ready evidence for privileged access, segregation of duties and change governance, mapped to frameworks such as SOX, SOC 2, ISO 27001, HIPAA and GDPR.

Visit xtype.io ↗

Book a demo

LinkedIn YouTube

Screens marked as sample data show xtype Compliance views populated with illustrative records; names and values are fictitious. ServiceNow, the ServiceNow logo, Now, Now Platform and other ServiceNow marks are trademarks and/or registered trademarks of ServiceNow, Inc. in the United States and/or other countries.