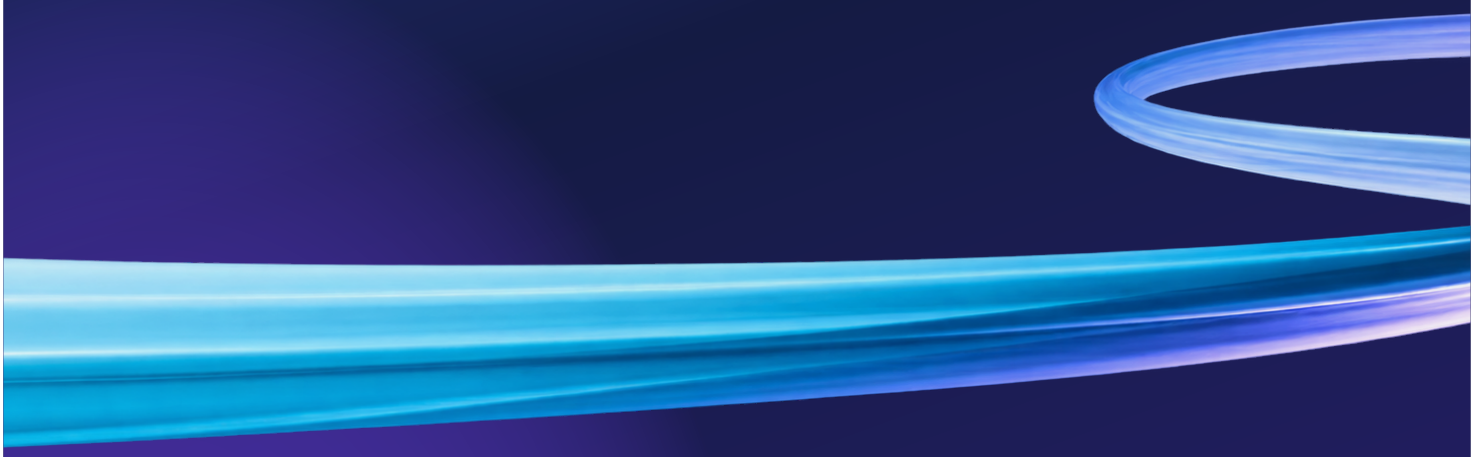
xtype | servicenow

AN XTYPE E-BOOK

Who Changed That Role?

Controlling identity and access changes across ServiceNow instances



WHAT'S INSIDE

Contents

	Introduction	3
1	The question that always comes	4
2	ServiceNow has good controls. They stop at the instance edge.	6
3	The anatomy of an access change	8
4	The four questions, and the record that answers them	10
5	Start this week, and let the record build itself	12
	Access Change Audit Checklist	15
	Let the record build itself	16
	About xtype	17

Product screens are from xtype Compliance for ServiceNow, shown with illustrative demo data.

INTRODUCTION

The question every auditor asks

Every auditor eventually asks the same question: who changed this access, when, and why? Most platform owners can answer for today. Very few can answer for last March.

This short guide explains why access changes in ServiceNow are so easy to miss, what an access change really looks like, and what kind of record lets you answer the question in minutes instead of days.

IN THIS GUIDE

Chapter 1 shows the four kinds of access change that nobody sees.

Chapter 2 explains where ServiceNow's strong native controls stop.

Chapter 3 maps the five ways access can change, and what counts as powerful.

Chapter 4 sets out the four questions an auditor asks, and the record that answers them.

Chapter 5 gives you three steps to start this week, and a one-page checklist.

01 CHAPTER 1

The question that always comes

Sooner or later, an auditor points at one account and asks a simple question: "Who gave this person this access, when, and why?"

Most platform owners can answer for today. They can export a list of roles in a few minutes. But the auditor is rarely asking about today. The question is about a date months ago, often on an instance nobody was watching closely.

The access may be correct. The problem is that nobody can prove it.

The team spends days rebuilding the answer from tickets, emails and memory, and the finding is written anyway.

The invisible-change problem

Most access changes are not hidden. They are simply never seen. Four kinds of change are especially easy to miss:

- **Changes outside production.** Access changes in development and test build up quietly, because reviews usually look only at production. But many problems start in those instances.
- **Changes that do not touch the person.** A person joins a group, or a role is added inside another role, or an access rule gains a new role. The person's power grows, but nothing on their own user record changes. A review of each person's roles will pass.
- **Changes that are already undone.** A role granted on Friday and removed on Sunday leaves the access list looking correct on Monday.
- **Changes hidden in normal activity.** A platform upgrade can make hundreds of legitimate access-rule changes in an hour. One unplanned change inside that burst is almost impossible to find by reading lists.

SCENARIO · THE WEEKEND ROLE

On Friday evening, an administrator grants themselves a powerful role to finish an urgent task. On Sunday night, they remove it again.

On Monday, the quarterly access review runs. Every account has exactly the roles it should have. The review passes.

Maybe the task was legitimate. Maybe it was not. Nobody can tell, because the only evidence is in what briefly existed, and nobody recorded it.

None of these changes needs to be secret to be invisible.

They only need to happen where nobody is looking, or leave no trace on the record people check.

The answer is not more careful reviews of today's access. It is a record of every change, as it happened.

02 CHAPTER 2

ServiceNow has good controls. They stop at the instance edge.

ServiceNow is not weak on access. Most platform teams already rely on strong native controls, and this guide assumes you do too:

- **Elevated roles.** security_admin must be switched on for each session, so the most sensitive power is not always active.
- **Change approvals.** Change requests send work through review and approval before release.
- **Audit tables.** Changes to audited records are written down, field by field.
- **Instance Security Center.** Each instance is measured against recommended security settings.

These controls work well at the moment of action, inside one instance. The gap is what comes after, when someone needs to prove what happened over a whole period. Three things make that hard.

Each instance keeps its own story

Development, test and production are separate by design. Access changes build up in every environment, and no instance holds the full picture.

History does not last forever

Some logs are removed automatically after a short time, and many teams limit how long audit data is kept to control table size. The record you need most is often the record that no longer exists.

The record is not organized around the question

Many of the facts may exist somewhere. But joining them into an answer (who, what, when, through which path) takes a skilled administrator, a script, and time.

“

Your stack is probably secure. The problem is that you cannot prove it.

BEFORE YOU READ ON

Words used in this guide

Instance: one copy of ServiceNow. Most companies have several, such as development, test and production.

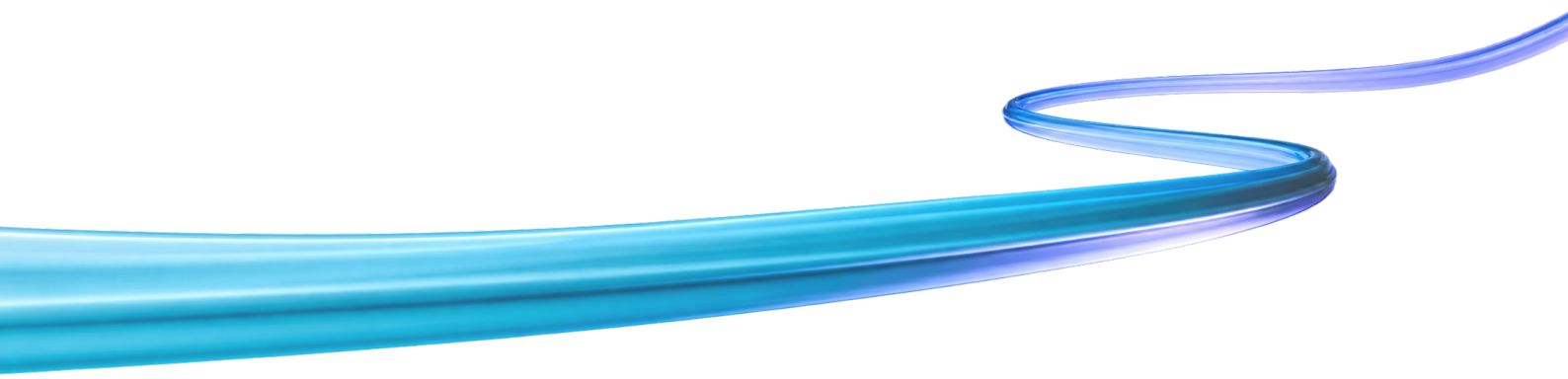
Role: a set of permissions. “admin” is the most powerful one.

Access rule (ACL): a rule that decides which roles may see or change a kind of record. An instance can have tens of thousands.

Stack: the collection of ServiceNow instances a company runs. Governed together, it becomes a single estate.

Group: a set of people. A group can hold roles, so everyone in it receives those roles.

Elevated role: a very powerful role that must be switched on for a session before it works.



03 CHAPTER 3

The anatomy of an access change

“Who has access” sounds like a list. In ServiceNow, it is closer to a map, because access can change in at least five ways:

1. **A direct grant.** Someone gives a role straight to a person.
2. **A group change.** Someone adds a person to a group, and the person receives every role the group holds.
3. **A role inside a role.** Someone adds one role to another, and everyone holding the outer role gains new power. This can go several layers deep.
4. **An access rule change.** Someone edits an access rule, or adds a role to it. What a role can do changes, but nobody’s roles change.
5. **An elevated session.** Someone switches on an elevated role for a short time.

A defensible record must cover all five. Most manual reviews only cover the first.

SCENARIO · THE GROUP THAT GREW

In January, a developer adds a powerful role to a support group so the team can fix one urgent problem. The fix works. Nobody removes the role.

Over the next six months, the service desk adds eleven new people to that group as part of normal onboarding. None of them asked for the powerful role. None of their user records show it was granted to them.

In July, an auditor finds twelve people with far more access than their jobs need. Nobody granted it to any of them directly. One change in January did it all.

What counts as “powerful”?

Do not judge a role by its name. Judge it by what it can do. A role is powerful if it can:

- change code that runs on the platform, such as scripts and business rules, including through delegated development, where people who are not administrators are allowed to build;
- change the access rules that decide what other roles can see and do;
- read or change sensitive data, such as HR or financial records;
- or contain another role that can do any of these.

Write the list down, agree it with security and compliance, and review it on a schedule. Many teams find powerful roles that are not called admin at all.

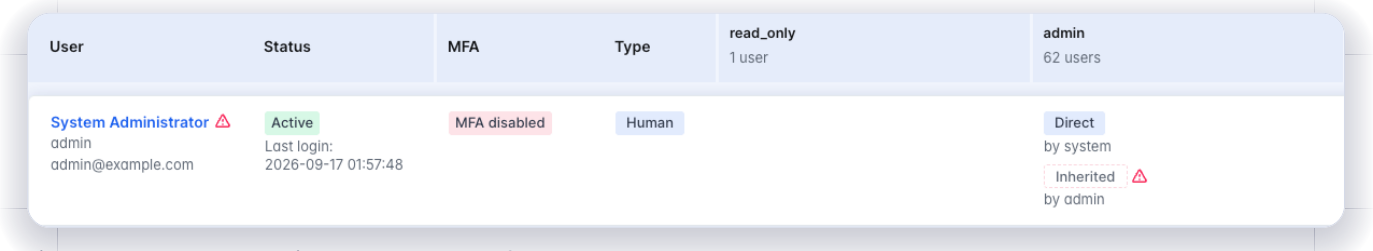


Figure 1: How a person received a role. One row of the Users with Elevated Roles report answering the question “how did they get it?”: this account holds admin twice, once given directly by system, and once inherited through a group from admin, flagged with a warning because the account granted it to itself. Illustrative demo data.

04 CHAPTER 4

The four questions, and the record that answers them

For any access change, an auditor wants four answers:

1. **Who made the change?** A named person, not only a service account.
2. **What exactly changed?** Which role, group or rule. For access rules, the value before and after.
3. **When did it happen?** An exact time, and a record that covers the whole audit period, not only today.
4. **Why, and was it reviewed?** A written reason, and a decision by someone other than the person who made the change.

If you can answer all four for any change in the period, most access findings disappear.

What a defensible record looks like

- **Every change is an event.** Each event names who, what and when. Comparing two exports only tells you that something changed, not who changed it.
- **Every grant shows its path.** Direct or through a group, and granted by whom.
- **It covers every instance.** Keep the record on development and test as well as production. Many problems begin before production.
- **It is honest about when it started.** A record with a clear start date is credible. No tool can recover history the platform already deleted, which is the best reason to start now.
- **It carries its own proof.** An export should state what period it covers, when it was made, which instance, which filters, and who made it.

SCENARIO · TWO TEAMS, ONE QUESTION

Two companies receive the same audit request: show every change to administrator access on production in the last quarter, and whether each one was reviewed.

The first team exports today's role list, searches old tickets, and asks colleagues what they remember. It takes four days, and three changes still have no explanation.

The second team opens a report for the quarter, filters it to the admin role, and exports it. Every grant and removal is listed with who, what and when. Each flagged change has a reviewer's decision and a written reason. It takes ten minutes.

Both companies may be equally secure. Only one of them can prove it.

05 CHAPTER 5

Start this week, and let the record build itself

Pick your production instance and do three things:

1. Write down which roles your organization treats as powerful, based on what they can do.
2. For each one, list who holds it and how they received it: directly, or through which group.
3. Try to answer the four questions for one change from three months ago, including one change outside production.

If step 3 takes longer than an afternoon, the gap is not your team. It is the record.

How xtype helps

xtype Compliance adds reports to xtype for ServiceNow that keep this record for you, starting on the day monitoring is switched on. Each report covers one instance, so you keep the record on every instance you choose. Compliance teams can read the reports without ServiceNow administrator rights.

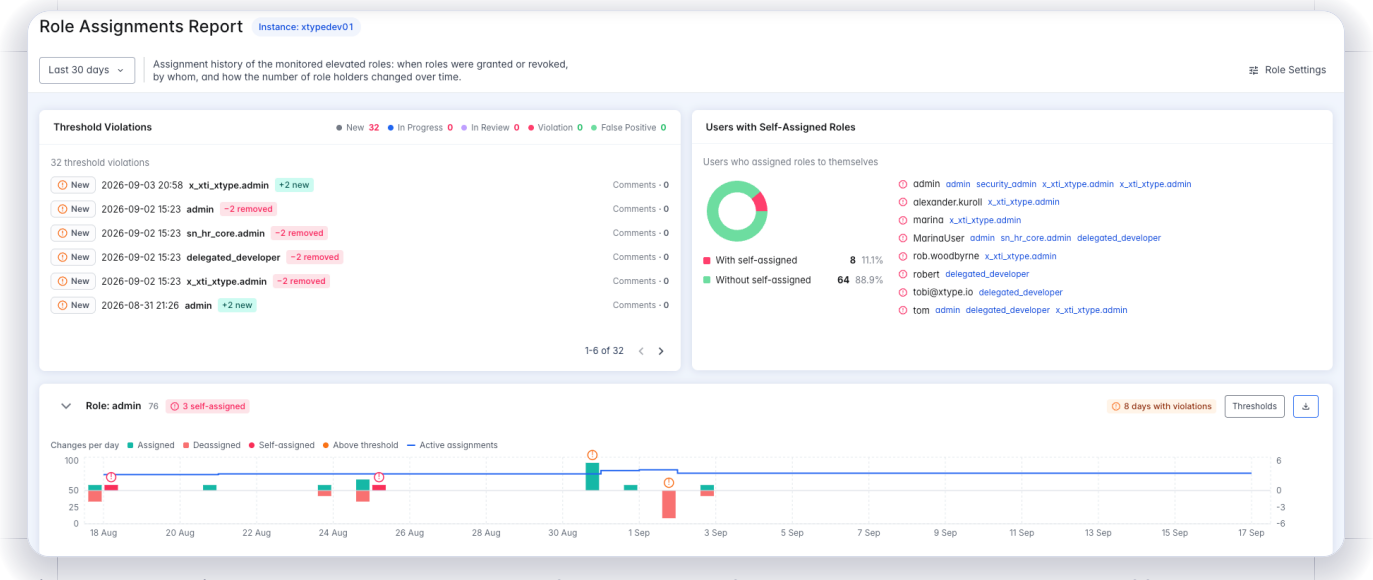


Figure 2: Role Assignments. Every grant and removal of a watched powerful role over time, who made it, and a list of flagged items to review. Here, 32 threshold violations are waiting for a decision, 8 users have assigned roles to themselves, and the admin chart shows each day's grants, removals and self-assignments. Illustrative demo data.

- **Role Assignments** records when selected powerful roles were granted or removed, and by whom. Simple rules flag unusual activity, such as a role granted and removed again within a day. That is the weekend role, made visible.
- **ACL Audit** records changes to access rules, including a role added to a rule. For each changed field, it shows the old and new value where ServiceNow audits that field. System upgrades and plugin installs are marked, so routine bursts explain themselves.
- **Direct Role Holders** finds roles given directly to people outside your group model. It checks groups inside groups and roles inside roles before it raises a finding.
- **Users with Elevated Roles** shows who holds powerful roles today, and for each one, whether it was given directly or inherited, and by whom.

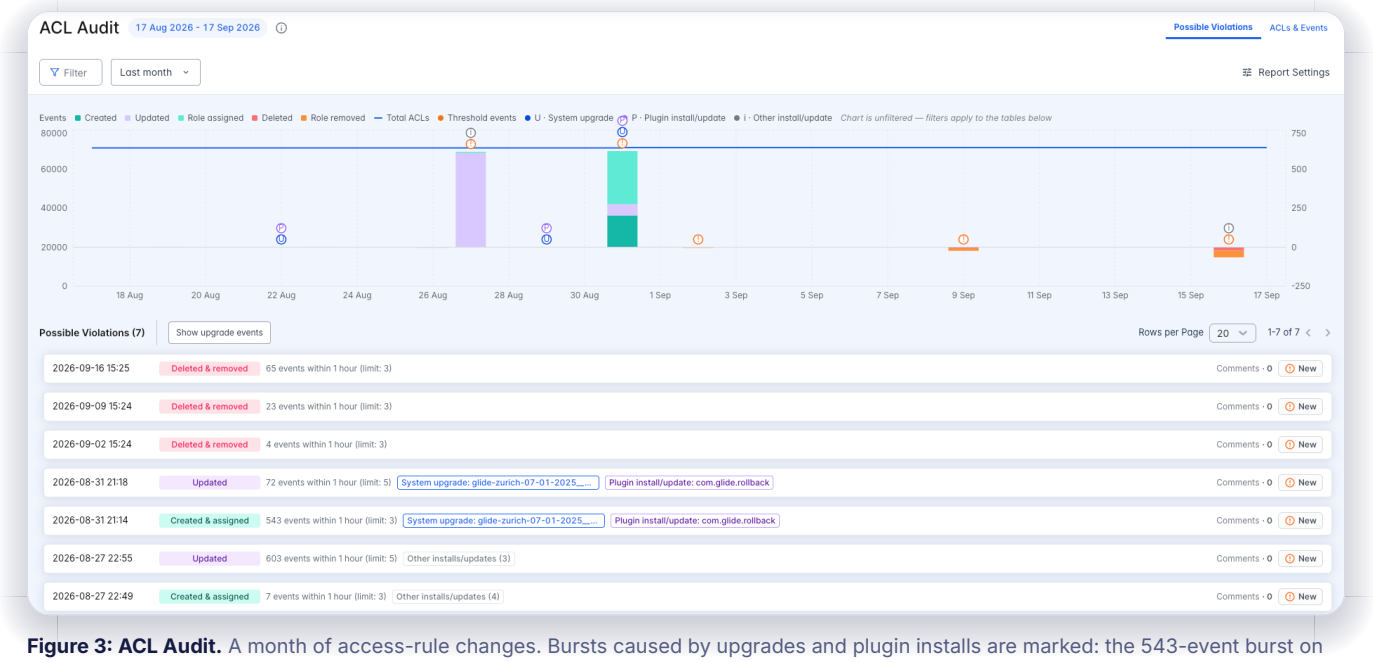


Figure 3: ACL Audit. A month of access-rule changes. Bursts caused by upgrades and plugin installs are marked: the 543-event burst on August 31 carries a system upgrade tag and a plugin install tag. Bursts with no mark, such as 65 deletions within one hour on September 16, stand out for review. Illustrative demo data.

CLOSING THE LOOP

Every flag gets a decision, every file explains itself

Reviewers can give each flagged change a status and a required written reason.

Every report exports to a spreadsheet with a built-in sheet that records the period covered, when the file was made, the instance (scope), the filters, and who exported it.

NEXT PAGE

The Access Change Audit Checklist for ServiceNow: 20 checks in five groups. Complete it once for each instance, starting with production. Go to the checklist →

CHECKLIST

Access Change Audit Checklist for ServiceNow

One page. Complete it once for each instance, starting with production.

Instance	Period reviewed	Reviewer

1. Know your powerful roles

- [ ] We have a written list of roles we treat as powerful.
- [ ] The list is based on what each role can do (scripts, business rules, access rules, sensitive data, roles inside roles), not on its name.
- [ ] Security and compliance agreed the list in the last 12 months.

2. Know every path to access

- [ ] For each powerful role, we know who holds it and whether it was given directly or through a group.
- [ ] We know which powerful roles are given directly to people outside our group model, and why.
- [ ] We know which roles contain other powerful roles, at every level.
- [ ] We know which access rules protect scripts and business rules, and which roles they allow.

3. Record every change

- [ ] Every grant or removal of a powerful role is recorded with who, what and when.
- [ ] Membership changes for groups that hold powerful roles are recorded.
- [ ] Access rule changes are recorded, including roles added to rules, with before and after values.
- [ ] We know the date our record starts on this instance.
- [ ] We keep the same record on development and test, not only production.

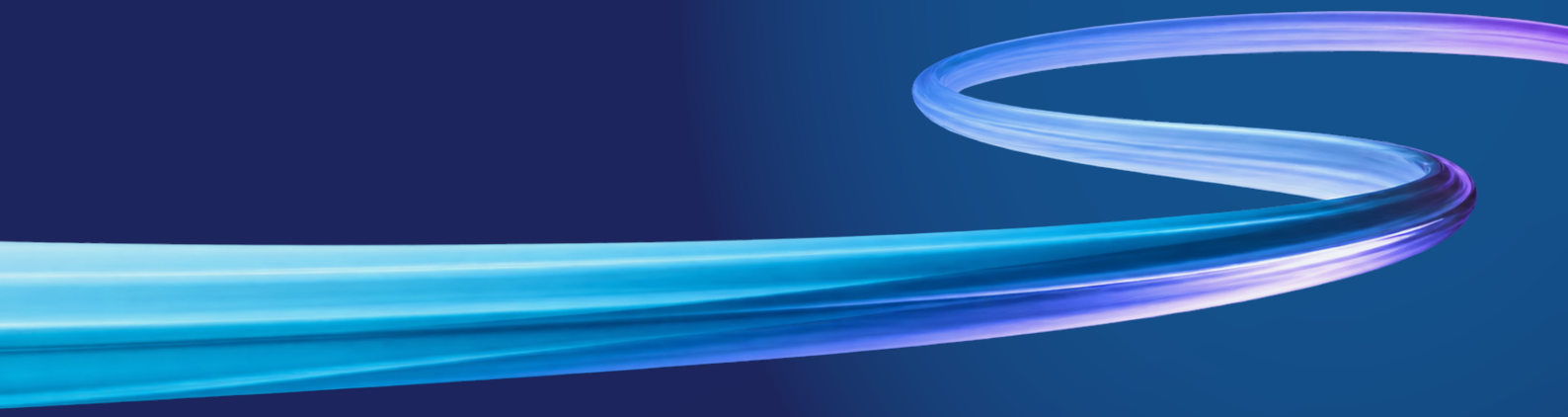
4. Review and explain

- [ ] Self-granted powerful roles in the period have a written reason.
- [ ] Roles granted and removed within a short time (for example, one day) were reviewed.
- [ ] Emergency access in the period was recorded and reviewed afterwards, with a written reason.
- [ ] Nobody reviewed their own change.

5. Prove it

- [ ] We can answer the four questions (who, what, when, why) for any change in the period.
- [ ] Our evidence file states the period covered, the date created, the instance, the filters and the author.
- [ ] Our retention period covers our audit period.
- [ ] We could answer the auditor's question for a date three months ago in under an hour.

Score: ___ of 20. Any unchecked box in sections 3 or 5 is where an audit finding is most likely.



LET THE RECORD BUILD ITSELF

Answer the question in minutes, not days

xtype Compliance keeps the record of every access change for you, on every instance you choose, starting on the day monitoring is switched on.

Book a demo →

xtype | servicenow

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 reports 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

Product screens show xtype Compliance with illustrative demo data. 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.