Engineer

Incident response

coordinating decisions and action during an incident

What it is

Incident response is the coordinated work of handling a disruption or a serious threat to a system. An incident is the event a team decides needs attention. The response is what the team does about it, from recognizing and declaring the incident to coordinating action, restoring a usable service and handing the remaining work to named people. It may involve engineering, operations, customer support and people with authority over legal, security or business decisions.

NIST's incident response recommendations for cybersecurity describe a cycle of detection, response and recovery supported by preparation and ongoing improvement. The publication is about cybersecurity. An engineering team can use related coordination practices for a broken release or a failed dependency, with the right people and safeguards for its own situation. The response to a suspected disclosure may require preserving evidence and protecting affected people even when the application is still serving traffic. In an ordinary outage the urgent choice may be how to restore that traffic without adding new failures.

Google's account of incident management describes explicit coordination roles. An incident commander keeps a view of the whole response, an operations lead coordinates changes to the service, and a communications lead keeps stakeholders informed. One person can cover several roles early on. As the work grows, naming who owns each function helps people avoid duplicating effort or giving competing instructions. The roles describe responsibilities a team can allocate to suit its size.

Use the concept in a decision

Put “What is incident response?” to work in a real decision. We can help relate the concept to the system you are building or buying. Monitoring and alerts that speak up while a fault is still small, and the figures that show how often it stays that way.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Built the company's first observability dashboard — the company's first view of its own systems

Built the company's first observability dashboard Get in touch →

Why it matters

The early minutes of a problem bring information and decisions at different speeds. A check can say that payment failures rose within a few seconds, while the team may need several minutes to learn which customers or payment methods are affected. Support may see a pattern before an engineer does. A shared timeline can record both observations with their sources and the confidence the team has in them. Other responders can then investigate without treating the first explanation as fact.

A response also gives the team control of changes made under pressure. A rollback and an emergency configuration change can each look sensible on their own and be harmful if they happen together. The person coordinating operations needs to know what others are about to change, when the result can be observed and what the reversal would be. Clear authority at that moment leaves room for a specialist to propose a solution without leaving the outcome to whoever reacts first.

Recovery and diagnosis run on different clocks. Redirecting traffic away from a failing instance may restore the service while the underlying fault remains unknown. Marking recovery ends the period of interruption using an agreed observation, such as successful orders reaching the expected volume and stalled orders being reconciled. The later postmortem can examine why the failure was possible, how well the response worked and what to change. Its conclusions are stronger when decision history and test evidence have been preserved during the response.

What it looks like in practice

Imagine that a booking service starts taking payment while sometimes failing to issue a ticket. Support sees reports first and engineering confirms that requests have started failing. The first responder declares an incident and records when the symptoms were first seen. The team agrees on a short description of the known effect and appoints someone to coordinate the service changes. Support has a point of contact and a time for the next update, even if the cause remains unknown.

Operations inspects a small sample of affected bookings and finds a recent change that might explain the timing. The coordinator asks the team to confirm that returning to the previous version leaves already captured payments visible. One person carries out that rollback and a second watches the relevant booking and payment indicators. Support preserves the identifiers for users who need follow-up. Customer communication reports the effect and advises what to do with an unconfirmed booking. The team can revise the message as it learns more.

The rollback is judged from complete transactions, including whether a payment and a ticket match, and from the progress of any backlog. The team records when the route became usable again and who will reconcile unsettled bookings. It writes down the decisions and observations needed for follow-up. The response closes when the current risk can be handed back to ordinary operations. The investigation into the cause can continue with an owner and a date.

Engineer's case on a macOS application's memory growth records separate observations of memory growth and the subsequent work to bound retained events. It illustrates why it is useful to preserve what each measurement actually covers. The booking service above is a hypothetical example of coordination and customer follow-up.

Where it goes wrong

The team waits for certainty before declaring. A probable impact can be investigated with explicit ownership and a scheduled update. A clear threshold helps responders recognize when coordination is worth starting. The team can stand down if later evidence shows the situation is less serious.

Everyone changes the system at once. Keep a short shared record of actions, including proposed changes and who will execute them. Observe the result of a change before introducing another where possible. When independent actions can really proceed together, make their boundaries clear.

Customers hear guesses. Report what users are likely to experience, any step they can safely take and when to expect another update. Explain uncertainty directly. An explanation of the cause can wait until the team has evidence that supports it.

The symptoms end and unfinished work disappears. Make the handoff explicit. A recovered service can still have a backlog, users who have not been contacted or a provisional safeguard that needs an owner. Put those tasks where the team will track them after the incident channel falls quiet.

Further reading

- NIST SP 800-61 Rev. 3 — incident response and cybersecurity risk management, including preparation and improvement.
- Managing incidents — Google's description of coordination roles, communications and an incident record.
- Incident response — incident-management experience and practical examples from Google's SRE workbook.



Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is incident response?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/incident-response/