How security incidents are different... and how they're exactly the same

Laura de Vesine
Alec Randazzo

Who we are

- 20+ years of incident response
  - Split between “SRE” and “security”
- Worked together at Datadog to rebuild security incident response
  - And responded to real incidents together
  - Also both have experience with incident response elsewhere

Laura: so, you heard a little bit of an intro already. I want to give a little bit more background on who we both are because so much of this talk is about what we already know and what we learned from each other. Between the two of us we have more than 20 years of incident response experience. Until recently we were working together at Datadog on an overall security incident response process, rebuilding a program that had gotten into some trouble. We also responded to real incidents together – in fact, we got to work together on one in Alec’s third week. In addition to the incident process at Datadog, we also both have practice and experience in incident response elsewhere, so we have at least some perspective on the industry overall.

Image public domain

Alec knows about security!

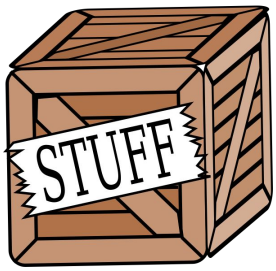
13+ years of security incident response, from routine (BEC) to complex (nation-state attacks on Fortune 500s). Believes Friday at 5 PM is cursed.

Alec:
Thanks Laura! Hey everyone! I'm Alec. I've worked in the security Detection and Response space for over 13+ years, most of which has been responding to security incidents. I've responded to incidents ranging from the mundane business email compromises to exciting nation state compromises. I'm currently a staff engineer on Datadog's Security Incident Response Team.

Personal photo. Image AI generated and public domain

Laura does reliability… stuff

- SRE & “reliability” incidents
- Focus on larger-scale orgs
- Security incidents ~3 years



Laura: okay so briefly hi, hello, I’m Laura. I’ve been in the SRE space for 10-ish years now, specializing in incident response and prevention. My background and introduction to incident handling as an SRE was at Google. I spent about 4 years leading incident handling org-wide at Datadog, and I’ve actually just started a new role at Reddit. I have a good sense of how larger-scale orgs handle their outages and incidents, what we as SREs think of as “best practices” in that space, and what I’ve seen work and not work. I first started working more directly on security incidents about 3 years ago, and it’s been a learning experience

Personal photo. Comic from https://reliamag.com/cartoons/shutdown-turnaround-outage-planning/ under CC4. Other images public domain.

“Incident” means something different to security

- “Incidents” often have regulatory requirements in security
  - Folks may be very careful about using the term
  - Still use the incident process for “events”

Laura: The very first thing I learned when I started working with folks in the security incident space is that “security incident” is a loaded term. As soon as you officially call something a “security incident” a lot of people care – compliance, company lawyers, very possibly the board and the SEC. Lots of things that I would handle with our incident response process (and therefore call an “incident”) don’t actually fall in that category – we’ve leaked a password somewhere; we have a new critical CVE to repair; even someone internally accessing something they shouldn’t have in many cases. For security folks <click>, for something to be an “incident” it generally needs to have an actual malicious actor verifiably gaining access to systems and data they should not have access to. That’s not to say that security professionals don’t use a very similar incident response process for security issues that don’t have that malicious component – in fact, they do – but they’re likely to be very sensitive to having them *called* “incidents”, because of that standard and the associated reporting requirements.

For this talk, we *are* going to be discussing an event that would fall under the classification of “security incident” for an org, but (for reasons we’ll also talk about), this is a made up event – *not* something that really happened. It’s useful to remember that (just like for a lot of SRE teams), your security response team(s)’ day-to-day is much more focused on lower-severity issues like “we logged an access key” or “someone installed malware and our endpoint monitoring immediately quarantined it” – and not “a bad guy is in our systems”. The process for responding to these will largely look and feel even more familiar to you as an SRE.

Obligatory Disclaimers

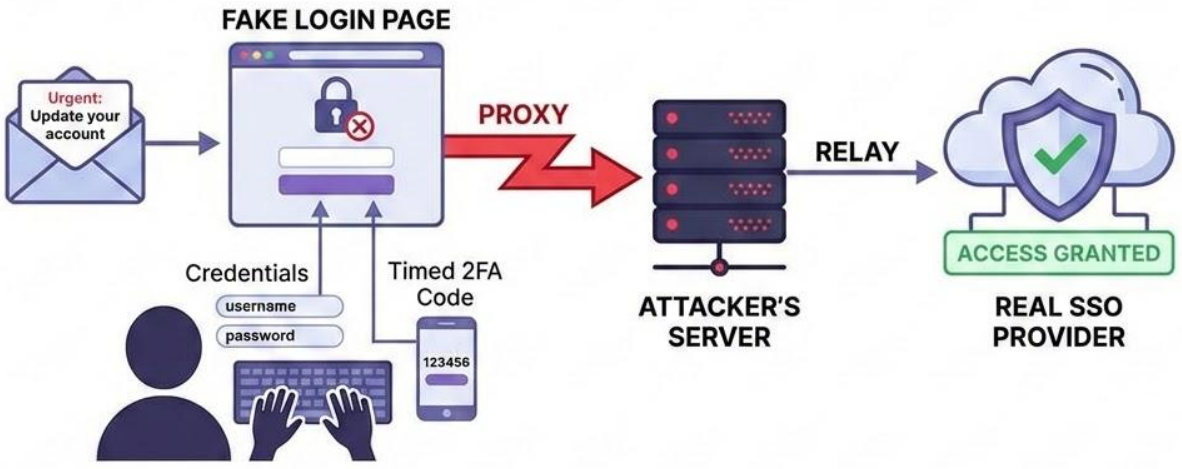
- The incident we will describe is **NOT** based on a true story
- We are **NOT** speaking about or for our employers
- We are **NOT** lawyers
- Do **NOT** interpret our presentation as legal advice

Alec
First some disclosures.
<click>
While the mock incident is realistic, it is not based on a true story. It's an embellished scenario inspired by experience, open source CISA intelligence, and vibes.
<click>
We are not speaking about or for our employers. We are speaking from personal experience across many different companies.
<click>
<click>
We are not lawyers and nothing in the presentation is meant to be legal advice. We we'll mention legal things which are based on observing actual lawyers responding to security incidents.
<click>
Image AI generated and public domain

Mock extortion incident

Alec:
I'll describe the full end to end scenario upfront to set the stage. It's a data theft extortion scenario which sadly is common nowadays.
<click>

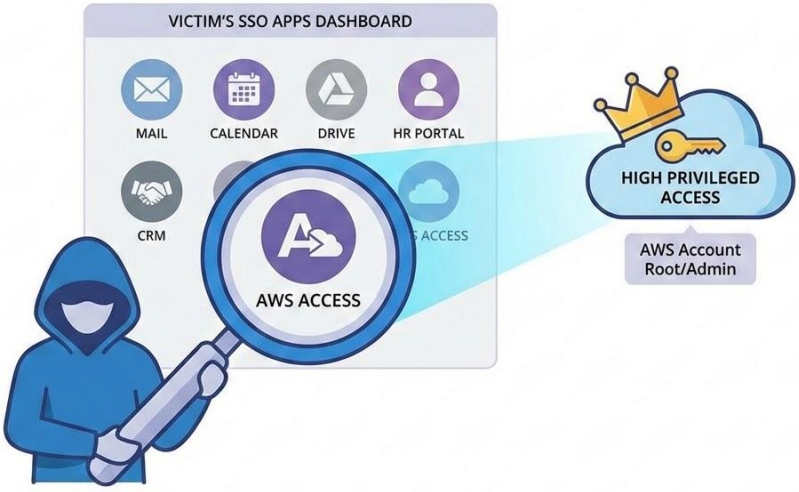
Step 1 - Gain access to SSO



Alec:
The threat actor's first step is to gain access to an organization's SSO. One of the most popular ways to do this is sending a phishing email to an employee thank links out to a look-a-like SSO login page. This site proxies the username + password to the actual SSO provider. When the SSO provider prompts for a 2FA code, the site proxies this request back to the user. The user inputs a 2FA code which the site proxies back to the SSO provider. Then the SSO provider provides the site with a session cookie, at which point the threat actor is in.
<click>

Image AI generated and public domain

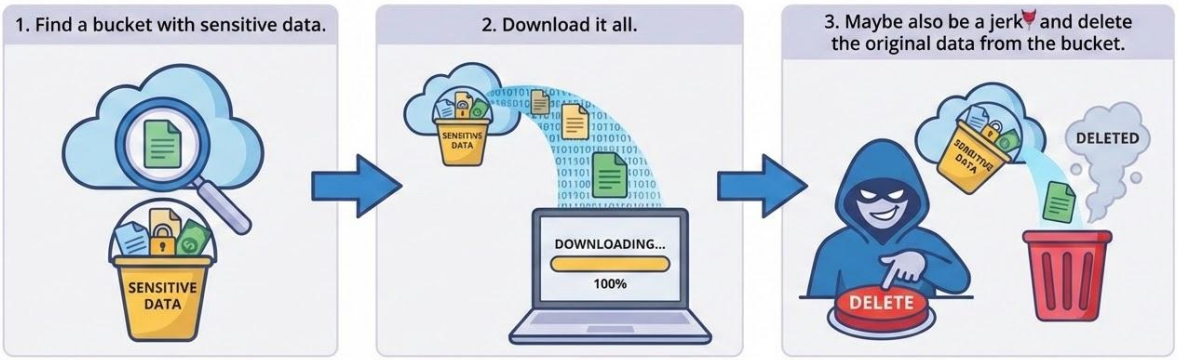
Step 2 - Explore



Alec:
Next, the threat actor will then explore what applications they have access to in SSO. They are specifically looking for apps that contain sensitive data. Let's say in this scenario the victim was an engineer with elevated privileges in their organization's AWS accounts.
<click>

Image AI generated and public domain

Step 3 - Steal data



Alec:
The threat actor then explores S3 buckets looking for sensitive data. Once they find some, they will download it and then possibly delete the bucket contents hoping they have the only copy.
<click>

Image AI generated and public domain

Step 4 - Extort

Alec:
They will then try to make contact with the victim organization with an extortion demand for crypto and a threat to publicly leak the data.
<click>

Image AI generated and public domain

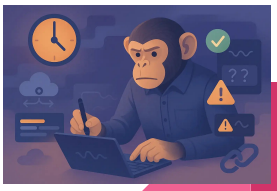
Step 5 - (maybe) Profit!

Alec:
If the victim organization pays the extortion then the threat actor profits!
<click>

Image AI generated and public domain

Detection: Monitoring and defense in depth

- Detect at exfil or first unauthorized use
  - Cut off access; done!
- User data access is limited
  - And admins are extra careful
- Worst case, we have immutable backups
  - That we've practiced restoring recently



Laura: Okay, so someone is in our systems. Obviously we have multiple security folks working on detection and response (just like we do for reliability questions). So we'll detect as soon as our employee's credentials are exfiltrated, or, at worst, as soon as someone tries to use them from an unauthorized system or to do something suspicious. Clearly we have accurate monitoring for all of that that pages on true positives with high reliability. <click> So *of course* we just cut off the attacker's access basically immediately, all done. <click>

If, for some reason, our monitoring fails, of course users have access only to the data they actually need to do their job, and can't just go delete or overwrite data without getting someone else involved – we do that for reliability reasons, to keep users from being able to fat-finger a command and make a terrible mess, to say nothing of our security needs. <click> A few users (like me, probably!) have a lot more access, but I'm very careful with my credentials and would never fall for some kind of social engineering attack. <click>

And as a reliability person I believe in defense-in-depth, so even if our monitoring somehow fails and the exploited credentials have way too much access, I, as an intrepid SRE, have made sure that we have immutable backups <click> and the ability to reliably restore from them <click> (pause for laughter).

Images AI generated

In reality…

- Probably learn at time of extortion
- Deviations to security posture happen
- Malicious deletion of backups sometimes not considered when deciding backup strategy

Alec
<click>
Unfortunately, companies often learn of the incident at the time of extortion. This is often due to companies not having monitoring for this specific type of incident, having only off the shelf monitoring tooling, or not having a well resourced dedicated detection engineering team.
<click>
Factors that make this incident possible are often rooted in deviations away from established security practices. There may be engineers or user roles that were given excessive privileges from a previous reliability incident that weren't rolled back. Or maybe there are documented or undocumented exceptions to security standards.
<click>
<click>
Also, backup standards are only as good as the scenarios considered and how well they are followed. Malicious deletion of backups is often not considered when deciding backup strategy.

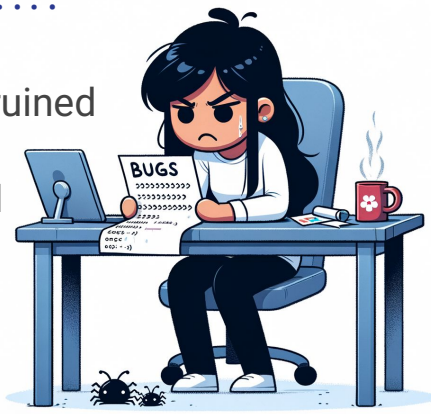
Ultimately, these incidents happen because the methodology these threat actors use broadly just works. They are motivated by profit and methodologies that don't work aren't profitable.
<click>

Image AI generated and public domain

So we detected… via demand letter?!?!

- My sadness is immense and my day is ruined

- Incident response 101
  - Triage
  - Mitigate
  - Resolve



Laura: Okay, deep breath, so we've detected a security breach by a real bad guy… because they sent us a demand letter. You know, I think that's even worse than detecting an outage through customer reports. …Alec, please tell me that at least the demand letter is going somewhere reasonable, and not to random salespeople on linkedin or twitter, right?
Alec: nope
Laura: [facepalm]. Well, I guess we're here now. <click>
Well, obviously this is a big incident. I'm going to spin up our incident process and start pulling in anyone I think can help – engineers who can tell us what data might be impacted, people who know how to shut down systems the attacker might be in and rotate credentials, anyone who can keep data from being deleted or further exfiltrated and otherwise stop the bleeding. Since this could affect pretty much anyone, we obviously want our response to be in public so we can get all the help we might need, and so people can report any impact they might be seeing in a proactive way. This is all incident response 101 and we're going to triage, mitigate, and resolve in a coordinated way: find out what data has been impacted, how it matters, and what the impact is (which includes validating that it was really accessed); halt unauthorized access and restore anything we need to; and then fix both the source of the breach and the ways our systems didn't stop the damage.

Images AI generated and public domain

Whoa there! Before we get too far into IR…

- Engage with Legal
- Do not respond to the extortion. There's no upside.

Alec:
Whoa there! Before we get to far in responding, we have some housekeeping that we have to do.
<click>
<click>
We need to engage with legal first. Legal has to be engaged with early so the investigation can be done at their direction for the purpose of providing the business with legal advice. This allows for communications and documents to be privileged which decreases the risk that they are discoverable in litigation.

To maintain legal privilege, the incident must be private and need-to-know only, lawyers must be on communications, and documents must have proper legal markings.

Legal privilege does NOT bury the incident. Facts can't be privileged. Privilege doesn't mitigate legally required notifications.
<click>
Additionally, do not respond to the extortion. There is no upside and lots of downsides. Engaging incentivizes the cybercriminal to continually increase pressure until they get payment. Engaging and not paying guarantees they follow through on their threats of a public leak. If you ignore them, there's a chance that they will cut

their losses and just move on.

If you pay them, there's no guarantee of closure. Cybercriminals often come back every couple of months to demand additional payments until an org decides to cut them off. Then the cybercriminal follows through on the original threat. So you paid for the outcome that you would have gotten if you paid nothing.
<click>

Image AI generated and public domain

Laura:
Alright, we have to make our incident private and get lawyers on everything. That's definitely going to slow us down, but we *are* still coordinating an incident, and the first thing we need to do is the same thing as we would for any incident still – triage and understand the damage. I happen to know our security org has extra big giant piles of logs (sometimes they ask SRE for resources to collect and store them even!). So we'll definitely have the logs. And security's tools for searching those logs are surely just as good as the tools I would use or better. I'm sure our experts even have a good idea of what to go looking for. That means triage will be easy!

Image public domain
Log size estimates taken from https://www.mobs-bd.org/siem-sizing-calculation-7000-seats-enterprise/ and other industry sources

Logging posture might be disappointing

- You may discover logging is a mess when you need it.
- Logs confirm the scope of an incident.

Alec:
<click>
<click>
I hope we have all logging turned on including the off by default S3 Data logging. I also hope we've been diligently centralizing all our logs into a platform with long retention that makes it easy to search for the off chance they you may need them one day. If not we still need to figure out a way to search them even if it's terabytes of logs sitting in S3. "We can't investigate because it's too hard/laborious" is not an option.
<click>
If we don't have logs, then there's a lot of downstream repercussions. We may not be able to prove what was and was not taken which means we must overestimate what was taken. We may not be able to identify which user account was compromised which means we'd need to rotate EVERYONE's credentials to ensure we kick out the threat actor.

Thankfully, in our mock scenario, we have all the logging we need.

<click>

Image AI generated and public domain

Block access; mitigation done!

- We just need to block the one user, which we can do cleanly without other impact
  - And that removes all access from our systems for the bad guy
- Restore any needed backups, some comms, and now we're ready to postmortem

Laura:
So naively, I would assume that we can block out the one compromised account, restore any needed backups, and do some basic communication about what happened based on our triage from the logs, and then we're all done with the main response and can move on to the postmortem. I guess that means that I'm assuming we have precise tools for cutting off that access that will affect only the bad guy and no one else, and that won't cause a larger outage in our own systems. Obviously for this incident with an account takeover, we'll need to block that employee's account, then either rotate their credentials and get them back in or make them a new account. If we were dealing with something like a DoS attack, we'd instead need precise tools for blocking attacker traffic. But either way it's quick and easy – block access once we find it, restore data, move on. Maybe it gets scary and complicated if the attacker is actually an employee, but then someone just fires them, right?

Image via wikipedia

Access blocked… we think

- Rotating credentials is only the first step.
- How did they the initially get in? Can they use that method again to get back in?
- Did they deploy backdoors onto systems, into source code, compromise *other* users, and/or create new users?

Alec:
<click>
So highest priority is indeed kicking out the bad guy. Kicking them out is not as simple as rotating SSO passwords and revoking SSO sessions for the one user and calling it a day.
<click>
You still need to figure out how they gained access in the first place because there's a chance that mechanism is still in place. Did they get access through a fake login proxy page or was it an infostealer infection? The former is a point in time theft of credentials while the latter is ongoing theft of newly generated session cookies.
<click>
You ALSO need to figure out if they added new ways to access the environment such as laterally moving and distributing backdoors on systems, backdooring source code, adding new users, and/or compromising other user accounts. If you don't remove every single way they can re-access your environment, they WILL come back if they want to. Depending on how sneaky they were, incidents could be a game of whack-a-mole for *months*.
<click>

Image AI generated and public domain

Now comes the real triage

- Security teams tend to be organized differently
- But delegating and coordinating incident work is universal

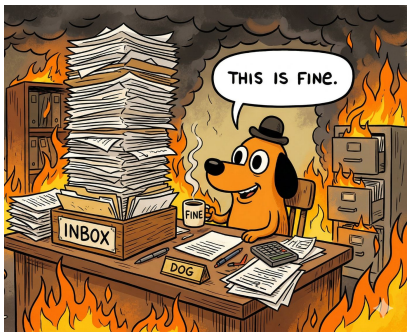
Laura:
This lateral movement thing – the fact that you're going to need to follow a single attacker through many different systems that might not normally be considered related – actually explains one of the biggest differences you see in security incident response *teams* vs SRE. SRE teams tend to be organized around specific systems and services, so that we can become experts in the systems we work with (and to give us the space and incentives to fix problems in the long run, not just mitigate as they happen). But security teams are dealing (sometimes) with intelligent bad guys, who are going to move around from your sales platform to your source code to your production systems if they can. If you have security response team boundaries between those things, you risk missing a movement and failing to resolve an incident because you didn't coordinate across that team boundary effectively.
But back to our incident – I'm in my happy place now. We have a bunch of things to do (explore a bunch of different systems for access and impact, lock off access as we find things). We can use all our standard incident coordination and communication techniques to track, organize, and delegate work. And while the incident is private, as long as I make sure people know that as they join, I can pull in whoever I need to help work on this incident, the same as I would for any major incident.

Gate image from
https://www.dreamstime.com/example-hyperart-thomasson-useless-structure-saigawa

-river-kanazawa-japan-case-gate-along-image289487627 by Tobyhoward
People talking image from https://pixy.org/

Get ready for the grind

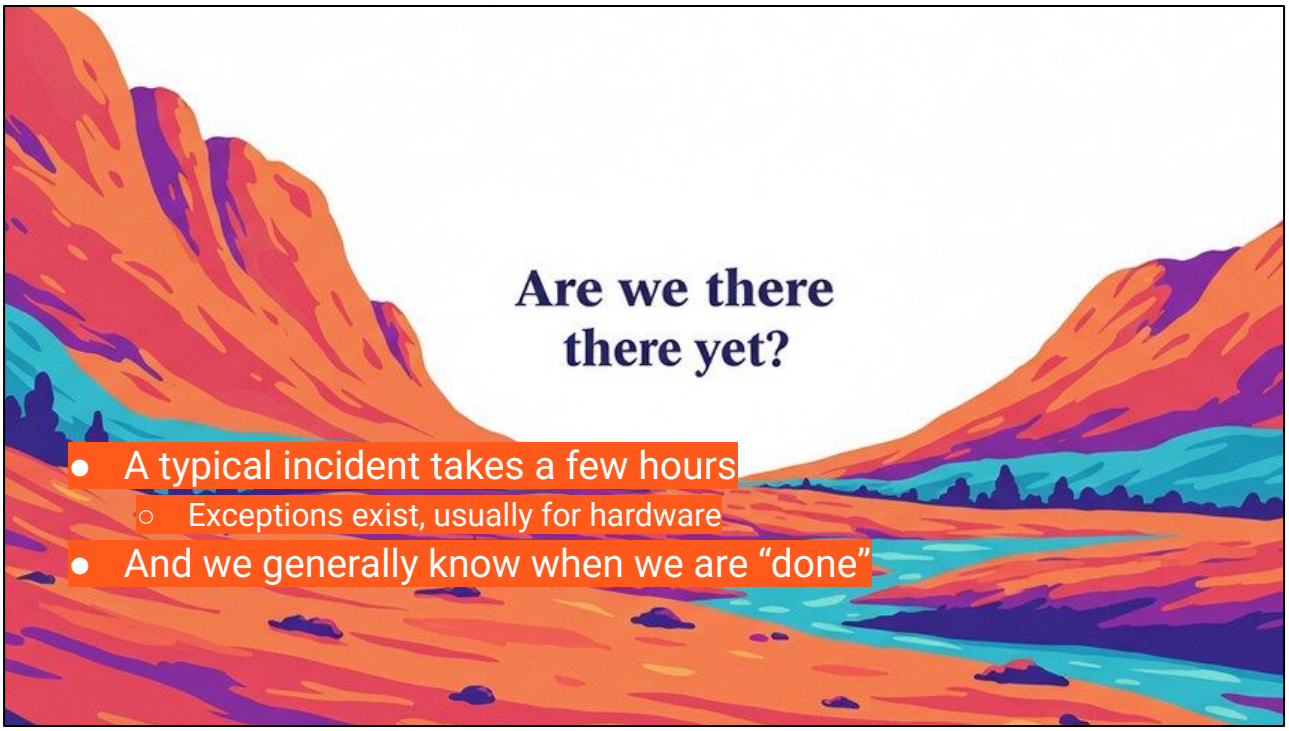
- Investigation scope can get absurdly large quickly.
- 1 SSO account touching 20 apps one of which gives ssh to 50 EC2 means you have 71 datasets to analyze.
- …and you still need to analyze the data they touched, no matter the volume.



Alec:
Organizing and delegating tasks is pretty similar between reliability and security incidents.
<click>
<click>
The main difference you'll see in security incidents is how quickly the investigation scope can grow to absurd levels. Every single system or app the cybercriminal touched requires a review to determine what they did.
<click>
If they used an SSO account to touch 20 apps one of which they use to get ssh access to 50 EC2s, then congratulation team we have 71 datasets to analyze. Level of effort of a cybercriminal versus level of effort for an incident response team is extremely asymmetric in favor of cybercriminals. Something that took them five minutes of effort may lead to hours of investigative effort for an incident response team.
<click>
Once you are confident that that the cybercriminal is fully kicked out, then there's the impact assessment where the team analyzes and categorizes data the cybercriminal touched, no matter the volume. Data discovery *sucks* because you need to do deep data analysis to determine if any data accessed is regulated by law, by contract, or if you are a publicly traded company, would have a material impact on company earnings. If there is such data, then there's a requirement to notify the individuals/companies whose regulated data was found in the datasets and possibly have to file an 81k with the SEC if you are a publicly traded company. If you are lucky,

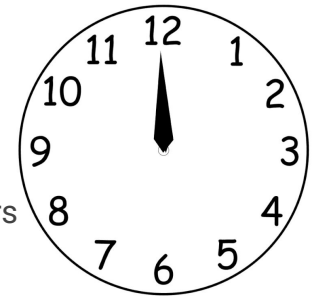
there is no regulated data and the incident was not material, which means the incident has no legal or contractual requirement to become public. Few companies will voluntarily talk about an incident publicly if there is no requirement to do so.
<click>

Image AI generated and public domain



Laura:
This is another case where I’ve learned security incidents have a very different flavor. In my non-security experience, incidents are generally over and done with (aside from the postmortem) in a few hours. Occasionally we’ll have a really bad one, or for teams who tend to deal more with hardware, especially networking, we might wait for delivery times, but those are more the exceptions than the rule. In my SRE hat, multi-day incidents are incredibly uncommon.
As SREs, we also generally know (more or less) when the incident itself is “done”. We’ll need to write a postmortem, and sometimes long-term fixes can be hard to scope, but the incident itself is over when the customer impact is and we have a reasonable belief that we’ve prevented an immediate relapse. Some incidents need extra analysis around customer-specific impact or similar, but we’ll usually do that outside of the emergency context as well.
Security incidents, on the other hand… can be a real slog.

Help, there's lawyers in my comms



- For a production outage, fast comms are best
  - Ideally <10-15 minutes for first message to customers
  - 30-60 minute cadence
  - Minimal approvals to support speed
- Transparent, too
  - Tell customers what impact we know as soon as we know it
  - May spend some time understanding individual customer impacts
- Lawyers do things… differently

Laura:
As we spend all this time figuring out exactly what happened and what was exposed, still on an incident footing, <click> I want to think about what we've been telling our customers. My instinct is to post a public message for customers <click> as soon as we get that ransom demand to let them know we have a problem and we're looking into it, maybe within 10 or 15 minutes. Then <click> for the first few hours to be posting what we know and what we're still looking into every hour or so at least.
<click> because this is what we normally do during incidents, I'm also used to the approvals for those kinds of posts being *relatively* minimal, otherwise we wouldn't be able to get them out at speed.
<click> this is implied in the speed, but we find with outages that it's important to be as transparent as possible with customers – <click> tell them what the impact to them is, as specifically as possible, as soon as we know, including giving them workarounds. <click> Depending on our specific business, we might also spend some of our time figuring out individual customer impacts right away to include those in our comms.
<click> But I have a feeling you're going to tell me that these lawyers in our incident won't let us do things that way.

Regulations and lawsuits are top of mind

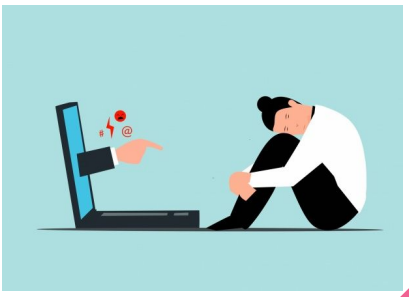
- Only disclose what you are legally or contractually obligated to.
- If the data is leaked, don't acknowledge the leak.
- This sucks, but it's the reality

Alec:
Your feelings are right!
<click>
<click>
I most often see lawyers advise against talking about an incident externally unless there's a legal or contractual requirement to do so. There's little upside in doing voluntary notifications. Sure, it feels good to be transparent, and there may be some customers that applaud the transparency and begin trusting you more, but there will be customers who will be upset. Maybe there will be lawsuits. It can turn into a self inflicted PR nightmare.
<click>
If the threat actor follows through on their threat to publicly leak data then it's often better to not publicly acknowledge it unless customers start asking questions about it. Security incidents are so common today that it frequently flies under everyone's radar despite a threat actor's best effort to make their work known publicly.
<click>
This sucks and I wish it wasn't this way, but this is the reality I've observed over 13 years across dozens of organizations.
<click>

Image AI generated and public domain

Postmortem time!

- Transparent internally, if not externally
- Sharing more externally builds customer trust
- Blameless



Laura:
Well, it's been weeks (or longer), but we've finally found all the data that was accessed, cleaned up and fully kicked the bad guy out, and we understand how they got in. We *finally* get to write a postmortem. Obviously this was a huge expense for us as a company and we never want to go through it again, so we want everyone to understand how and why it happened, and write a really transparent and comprehensive internal postmortem about what went wrong and how our systems can be built better to prevent this. We also want to examine how our response might have been better, either to detect the issue sooner (remember, we found out from a demand letter!), or to improve our ability to investigate and mitigate more quickly. Especially since any system *could* be a point of entry, sharing broadly will get us the best results.
Externally, I know the lawyers aren't going to let us share everything, but it builds trust with customers to explain what went wrong, make it clear that we have the tools and understanding to make it better, and that we're going to get things fully fixed. Being able to learn from incidents across the industry with shared transparency makes us all better. Sometimes public postmortems are a little high level, but most of us have enough background knowledge that we can get a pretty good picture of what kind of failure happened and what's going to be done to fix it.
And finally, obviously our postmortem is blameless – we're not here to focus on which humans did what wrong, but rather on how we can make sure our system doesn't

have this vulnerability anymore.

Cracked screen photo from https://www.flickr.com/photos/afdn/249821991/
Other images public domain

Remember those lawyers?

- They prefer nothing in writing
- Post mortems often *do* get written but with lots of restrictions and wordsmithing requirements.
- Remember legal privilege? Time to put it to use!

Alec:
<click>
<click>
Post mortems are still blameless in security incidents. While lawyers prefer that nothing is in writing, post mortems are high value so they do get written.

However since the document reflects on what went wrong, why they went wrong, and what security controls are missing, it is GREAT material for a lawsuit. If there's a lawsuit, this document is great to use the organization's own words against them to try and show negligence. We'll hedge against this possibility by being careful with wording we use in a post mortem.
<click>
This is also where legal privilege can help reduce the risk of the post mortem being in scope of discovery. Though legal privilege means access to the post mortem is on a need to know basis, even internally.
<click>

Finally, we fix

- Incidents may spawn large projects
  - Teams responsible for work have context
  - And understand priorities
  - Dashboards!

Laura:
Wow, I really hate that we can't even share internally – that makes me really worried we're going to have the same problems happen again. And as a person who uses computers, knowing that I'll only hear about someone's security problems if they are legally required to tell me about them doesn't make me sleep well at night either.
But okay, we do want to fix what we can. From my own experience, the postmortem is going to identify actions for various teams. Some of them will be fast, some of them might be year or more long projects, which can be reasonable to take on if the failure is large enough. Every organization struggles to prioritize these actions in the best way, but teams who have those actions will understand what problems they are trying to fix and why they matter, so they'll be able to advocate for reasonable prioritization and when they do the work they'll understand whether they've solved the underlying problem in the best possible way. I would also expect, for a large enough incident, that someone makes a public dashboard (or more than one) for executives and stakeholders to track progress on various work.

Images public domain

You may not know why fixing something is a priority

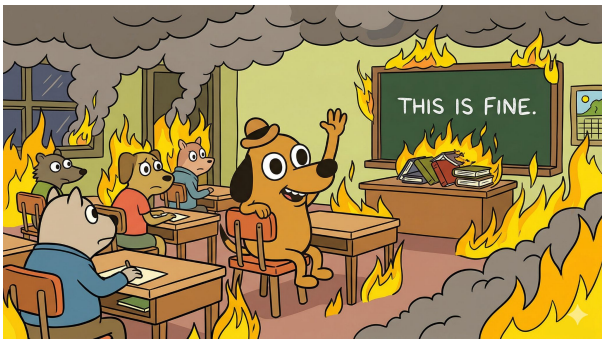
- Legal privilege ties security's hands regarding sharing details on why a mid-quarter drop-everything-else priority comes up.
- Purpose of the changes is to prevent a repeat incident

Alec:
<click>
<click>
Dashboards, widely viewable tickets, and updates to leadership also happen after security incidents. It's just that these don't go into much detail on the *why* these actions need doing to maintain legal privilege.
<click>
Please know that the asks are coming from a good place. Everyone involved in the incident including leadership, security, legal, etc want to make sure the incident doesn't happen again. These kinds of incidents at best cause burn out or at worst are traumatic for the folks involved. No one wants to go through the process again.
<click>

Image AI generated and public domain

How can you help during your next security incident?

- Abide by rules of privilege
- Ask before taking action
- Ask if there's tasks you can help with
- Call out valuable log sources that show user/admin actions
- If the incident commander is being calm, match the vibe.



Alec:
<click>
If you find yourself in the middle of a fire in security incident land, here's some tips.
<click>
Abide by the rules of privilege. It's possible to single handedly ruin legal privilege for the entire investigation. Generally, this means only talking about the incident with individuals already involved with the incident, putting special markings on documents, and including legal in all email, slack, and voice discussion about the incident. When it doubt, ask for guidance from the legal representative involved with the incident.
<click>
Ask before taking an action. Nuking a system that infected with malware without first collecting evidence may lead to investigative question that cannot be answered.
<click>
Ask if there's anything you can help with. If the incident is big, there's probably more tasks than there are people. If you are given a task, ask "what investigative questions do you want answered?" and then focus on using the data available to answer those questions. If you can't answer a question or can't give a confident answer, keep notes on why.
<click>
Call out log sources that might be valuable. Security will know the typical log sources like cloudtrail. They will have less insight into in-house application logs.

<click>
If the incident commander is being calm, match their vibe. An incident commander being calm is a sign of experience. They might have seen this incident before and know what to expect and have predictions on whats the actual final impact. Or maybe they are being calm to make sure everyone else stays calm, because panicked people make mistakes.
<click>

Image AI generated and public domain

Can we plan ahead?

- Help with logs?
- Make friends!
- Unify processes

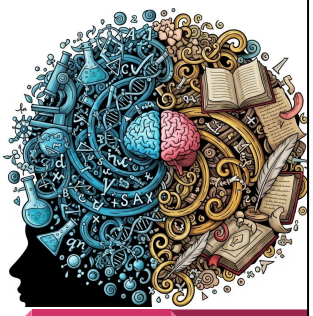
Laura:
So there's some things we can bring from the SRE experience that can help a lot with security incident response.
Firstly, you'll notice we talked about having large volumes of logs that may turn out not to be right when we actually need them – and might be painful to search and process, as well. At a minimum, can we make sure that we're only collecting data once and not duplicating it when the information we need is already in our monitoring? But in addition to that, production-ready pipelines for large scale log processing is an SRE happy place! It's quite likely your security team would appreciate your help here.
As SREs, we have a lot of touch points with many systems and teams in production. We can help our security teams build accurate mental models of what kinds of attacks are more likely to work in our systems, which can let them prioritize the right monitoring and response tools. We can also get involved with our security teams' chaos testing (they'll usually call them "red team" or "purple team" tests or exercises) to build relationships that security responders will be able to draw on when something goes wrong.
You, personally, can make friends with (some of) the lawyers who might be involved in security incident response. If those lawyers already know and trust you, they're a lot more likely to be able to successfully explain their concerns as part of the response, and to hear and understand what you need and can provide them in the moment. You should also make sure that those lawyers have the same expectations about

responsiveness and responsibility during an emergency as you do – do they understand “answer the pager in 10 minutes at 3am with the person who can actually make decisions”, or… not so much?
Finally, I really recommend advocating for a shared response process and tooling for your security incidents with whatever your “normal” incident response process is. You may get some pushback here – some security teams may not be fully aware that our response to outages is an organized, coordinated process with appropriate, clear responsibilities and data tracking. Invite them to join some existing incidents your team is involved with, and be open to hearing where they do have distinct needs (like having a tool that allows some incidents to be private to maintain privilege). But because your security team is likely to need to rely on other engineers for incident response, de-mystifying the process and ensuring that the response is mostly uniform regardless of the kind of incident you’re having will make your responses better overall.

Images public domain

Your skills transfer, except for the lawyers

- Security incident response is just thorough log analysis
  - Your “production sense” is a valuable asset
- Three things I use a lot in a security incident
  - What could I do with this data/access?
  - What would that exposure mean for customers?
  - Basic incident planning: next goal(s); how we get there
- Be careful not to “oversell” how bad a problem is
- Our incidents follow the same framework



Laura:
On the flip side, I want to encourage *you* to try and get more involved with your security response team(s). Aside from the lawyers, which for me at least was a new thing, <click> most of security incident response is actually something you already know how to do: thorough log analysis, with <click> your common sense about what’s important in production and how it’s put together guiding your intent and conclusions. Being able to bring your own, well developed, sense of what works and what doesn’t is a huge benefit to your security teams. When I joined security incidents, I found myself <click> using 3 skills most often. First, <click>, just figuring out what I could conceivably do (and *not* do) with the access or information that we thought an attacker might have. Second, <click>, thinking about what that would mean for our customers, including *why* someone might want to take actions we’d identified as probably working. Roughly speaking, your security team may call these two things together a “threat model”. Finally <click>, just basic incident planning and coordination: what do we need to do next, what will let us get there?
<click> one thing I had to unlearn was a certain level of catastrophizing about security incidents. The fact that someone could conceivably do something with *my* level of knowledge doesn’t mean that an outside attacker will, or did. The systems we work in tend to be pretty complicated, and while “security through obscurity” isn’t a best practice, we get a surprising amount of it just through the complexity of our environments. I find that it can help to remember that security is “broken all the time”

the same way that production is “broken all the time”. Discovering a problem that could cause a future outage, or even having a minor incident, aren’t cause for panic – just fixing the problem in our normal, professional, appropriately prioritized way. And sometimes we find potential outages where 12 things have to go wrong at once for the system to actually go down – we certainly like to fix those, but we think about likelihood in our prioritization. The same thing should apply to your thinking about security.
<click> Our incidents follow the same overall framework, and de-mystifying the security side is mostly a matter of applying your common sense (once you get used to the lawyers)

Running man public domain; brain image AI-generated

Want to get into security incident response?

- Check out CISA advisories for cybercrime groups: cisa.gov
- MITRE ATT&CK for cybercrime tactics and techniques: attack.mitre.org
- Ask to get more involved with security incidents. Real world experience is the biggest differentiator in the market

Alec
<click>
Want to get into security incident response?
<click>
Check out CISA advisories on threat actor groups. Their advisories often give the play by play for how these groups operate including details on the specific tools they use and what their attacks look like.
<click>
MITRE ATT&CK for cybercrime tactics and techniques. This framework captures the majority of things these groups do to achieve their objectives.
<click>
Ask to get more involved with security incidents. Real world experience is the biggest differentiator in the market. And with that's let's get into Q&A
<click>

Image AI generated and public domain

Q&A