Season Four Episode 1 | The One With Security and Jessica Theodat

[JAVI BELTRAN, "TELEBOT"]

STEVE MCGHEE: Hi, everyone. Welcome to season four of The Prodcast, Google's podcast about site reliability engineering and production software. I'm your host, Steve McGhee. This season, our theme is Friends and Trends. It's all about what's coming up in the SRE space, from new technology to modernizing processes. And of course, the most important part is the friends we made along the way. So happy listening, and remember, hope is not a strategy.

—

JORDAN GREENBERG: Welcome to this episode of the Prodcast, Google's SRE production podcast. And let me say, we have a guest who's back, back, back, back, back again from a previous season, Jessica Theodat. Jess, hello?

JESSICA THEODAT: Hello, Jordan and Steve. Thanks for having me back.

STEVE MCGHEE: So Jess, you have a foot in two different worlds, I believe-- security and reliability, everyone's favorite two worlds. These are the things that keep things going. Am I right about that or how would you better describe-- how would you better intro yourself than my silly analogy that we just did?

JESSICA THEODAT: Yeah. No, for sure. So, definitely in the two worlds. I guess one way to pitch it is that I'm a security focused reliability engineer, and to break that down-- so on-prem, so I primarily focus on infrastructure and systems for Google's premises, so that includes things like data centers, and offices, locals, colos. And in terms of responsibility, my role is to identify and assess and mitigate potential problems. And it's not strictly scoped to just security, I take a look at the bigger picture. I look at the potential effects of the bigger picture. And we'll dive into that a little more later, hopefully.

So when it comes to risk management, it's not a one-size-fits-all. And so, fundamentally, the translation of security risks and reliability risk to business impact is highly contextual. And so the idea is that I help stakeholders and their teams understand what those risks are to their

businesses, help them prioritize it, and implement strategies to address.

STEVE MCGHEE: Nice.

JORDAN GREENBERG: OK.

STEVE MCGHEE: I thought it was funny that you said on-prem. Because I know when people think of-- people who work at Google and then they think of the word on-prem, they think of the stuff that isn't at Google. They think about their data center. But you're talking about the Google data center, which is the cloud on-prem, which is the stuff that the cloud actually runs on, which I think is pretty cool. So for customers of cloud, Google has an on-prem also. It's weird. But it's the thing that all your stuff runs on. So that's a good point.

JORDAN GREENBERG: And we have the champion here.

STEVE MCGHEE: That's it.

JORDAN GREENBERG: Is what I'm hearing.

STEVE MCGHEE: Yeah, for sure.

JORDAN GREENBERG: So SREs often claim reliability is the number one feature of any system. Is security feature number zero or where does that fall in the hierarchy now?

STEVE MCGHEE: Takes welcome.

JESSICA THEODAT: Oh, for sure. For sure. So interestingly, people often joke about this. But it doesn't actually reflect reality. I think what I found having both sides of the fence now and at the intersection, the two domains often pull each other in opposite directions. You have on one hand, where security wants to resist and control, and then you have on the other hand, where reliability wants to ensure availability. And what we're seeing-- I think that we all see, really-- is that most orgs struggle with this tension, and they usually overindex for one at the expense of another. And I think that the real challenge isn't ranking them, like which one is higher than the other, but rather actually finding the right balance between the two for your specific context.

STEVE MCGHEE: As an SRE, I've been focused on reliability and availability and latency and all

that stuff forever. And I've found that a good interaction with the security team is that we're both aiming for the same big goal, which is, we want to protect the user. We want to maintain the service level of the thing. And we want the service to actually be the right thing. We don't want it to be altered by someone else or whatever. And they're actually-- it's all in the same purpose. We're on the same team. But I've seen it work poorly on the other side where yeah, we're working on the same team, but we have our blinders on. We're only focusing on our thing. We're not being holistic about it.

In cases like this, you can see where teams are trying to roll back a broken change or something like that, but they're not able to because of a security control. And then when they say, oh, we need to go past the security control to allow us to fix the thing, there isn't a way. And "security says no." This is a common response.

We want to be able to roll out a security patch very fast bypassing the reliability concerns. We want to be able to push this patch to all the VMs at once right now, because this is really important because there's a thing out there. And the reliability team is like, no, you can't possibly do that. Same deal.

So I think being aware of this alignment is really, really important. And being able to think about this ahead of time and being able to say, same team everybody. Let's go here. Does that resonate with you?

JESSICA THEODAT: Absolutely.

JORDAN GREENBERG: Yes.

JESSICA THEODAT: Absolutely. Absolutely. And I think you really nailed it with that one. Because the interesting thing about the two spaces is that, once you find yourself in a position where you have an incident and you need to compensate for something, they're both very difficult to bolt on. It's very difficult to then try to make your system more reliable if you didn't consider it in the early stages of your design. Right?

JORDAN GREENBERG: Yes.

JESSICA THEODAT: And the same is true of security as well. It's very difficult to make

something more secure without having to necessarily redesign or refactor your entire infrastructure, which I've had experience with. And I will say it is a lot of heavy lifting.

JORDAN GREENBERG: Oh, you don't think this is fun to have to refactor a whole thing?

JESSICA THEODAT: Fun.

JORDAN GREENBERG: Yes. Fun.

JESSICA THEODAT: And so these are the things that we want to consider early on in the design phase of our systems. When we are approaching building different things that we're tacking on different feature, what we want to do is essentially work with this integrated thinking where you have and you're considering the perspectives of both security and reliability, and you're optimizing for the two spaces.

And so part of that involves understanding what the trade-offs are and optimizing for those things, again, with an integrated mindset. And the benefit of doing all of this is that what ends up happening is you end up with a system that users implicitly trust. Because ultimately, users don't care about your security posture, and they don't care about your reliability metrics.

STEVE MCGHEE: No?

JORDAN GREENBERG: No.

JESSICA THEODAT: I'm sorry, Steve, they just don't. They just don't.

STEVE MCGHEE: Bummer.

JESSICA THEODAT: But what they really care about is that the system works and that it works securely.

STEVE MCGHEE: This reminds me of our last episode actually. So considering risk in a system and generally when we talk about risk in SRE, we're talking about availability, and latency, and all these other things, and scalability, and blah, blah, blah. Clearly, security is a pool of risk as well. So being able to think about just risk in an abstract way to say, all the risks need to be looked at and considered and ideally mitigated and prevented over time.

And regardless of which side of the fence or the-- what did we call it, having one foot on each side of-- do you put feet on sides of fences? I'm not sure if that's possible. But being able to consider risk on both sides of this is hard to do in an organization that has two teams.

So having someone-- honestly like you, Jess-- that can be the interface between those two teams or at least say, hey, do this other team exists is, I think, super important. So I've seen teams where they've had people just be able to move between the teams. That, I think, is really helpful for groups out there in the world who want to try doing this about.

Just make sure you can have an engineer who was in reliability go into security or the other way around and just bring their understanding with them. And most likely, through some process of osmosis, you'll get a little bit more awareness of each side of this fence or whatever we're calling it. Yeah.

JORDAN GREENBERG: And lean on your TPM too if there is someone who is acting as connective tissue. Not to beat my own drum, but it sounds like something that's like, oh, did you know we have a team that does this already? Might be just the thing that you need to integrate into your system.

STEVE MCGHEE: So given all that, how do you suggest? What have you seen in terms of these two teams or these two worlds or hats or whatever you want to call it? How is the best way to get them to interact? Because I know it must be different from a startup versus a gigantic megacorp. Are there themes that can help these different teams interact with each other or maybe things to avoid or something like that? What do you think, Jess?

JESSICA THEODAT: So, yeah, I think you bring up a very interesting point. I find that it does work differently in small startups or startups and small businesses versus big companies. In a small company or a startup, what you might usually see is that the same people who are handling security are the same people who are handling reliability. And the advantage of that is that you have folks who are essentially considering the same domains, or both domains, rather. And because of that, that practice will most likely end up being part of your engineering culture, rather than them operating as separate functions.

Whereas with larger companies, what will usually end up seeing is two different organizations,

and the two have to be very intentional about their connection points, making sure they have joint planning, they're sharing metrics, and basically have visibility into each other's goals. And one of the ways that teams exercise this is by implementing a review process that evaluates both security and reliability impacts simultaneously.

So going back to the same original idea, when we're approaching a system design, we're taking into consideration both security and reliability risks, and we're optimizing for both. And part of that process is also evaluating what the impact of those optimizations for are like, both the impact of the optimization for security and for reliability.

And then the other thing that I've seen-- and Google has gotten, I think, really good about this-- is about developing shared tooling so that we are eventually looking at things with the same lens, or through the same portal at least anyway. So using things like automation frameworks, observability platforms, we're looking at the same information-- we may process them differently because we're working in different contexts. But more or less, we're working with the same set of tools and information available to us.

STEVE MCGHEE: Is that not true everywhere? Is this a hard to do thing, Jess? Is that what you're suggesting?

JESSICA THEODAT: Very hard.

STEVE MCGHEE: Oh, no.

JESSICA THEODAT: But it's not impossible. I think, again, just it ultimately just goes back to being intentional and understanding what are your business's goals, and how do you shape your team and your cultures to speak to those things? I don't know, Steve. What do you think? What have you seen?

STEVE MCGHEE: So I've seen-- outside of Google I've seen very distinct roles and very distinct tools and the lack of ability to even see the other team's tools, let alone use them or understand them. Very common-- very, very common. Inside of Google, I've seen a fair amount of overlap. An example that I like to share with customers that I talk to is when you're doing a security-- think of it as a patch, not necessarily a traditional patch, but a security update-- using the same

rollout mechanism that you would use for product changes as opposed to some other thing, like some security specific rollout thingy. That's great. I think that's really good. That said, that rollout thingy needs to now know about security mode potentially. So it might be able to need to the ability to apply itself to more than one shard at a time, or whatever it is. So this requires that the teams actually listen to each other in terms of we have this requirement for that tool, else we have to go build our own tool over there and it's not going to be as good, blah, blah, blah, blah, blah.

So having a constraint system that allows you to say you're in this mode now and you're in this mode now is a lot of software engineering, it turns out. Given all that, this I think leads us into our next point, which is what about when we're doing incident response, and it's a security incident, not a reliability incident? Again, similar words like probably similar tools maybe, maybe you're using the same kind of-- inside of Google we have a tool called OMG which is the best name for a tool like this, obviously.

JESSICA THEODAT: Yes, definitely.

STEVE MCGHEE: But what the security teams do is different from what the reliability teams do, I think. Tell me if I'm wrong. But I know that there's a lot more siloization of need to know information. So for example, when I see a security incident, most of the time it has a funny name that is clearly auto-generated by some program somewhere. The name has nothing to do with the actual outage. And that's, like I've heard, on purpose. Also, like I can't see most of what's in that OMG. Again, it's probably not because they don't like me. It's probably because of some actual reason. So what's going on here? Why does this happen? And is this a good idea? I thought we just talked about sharing information and stuff. Why do we have to have all of this fancy secrecy stuff? Is this real, Jess? What's going on?

JESSICA THEODAT: Yeah. So the sharing everything with everyone approach works for reliability, but that can actually harm your security response. And the reason for that is that threat actors are always listening and watching for your response. And sharing that information can tip them off. So compartmentalizing information isn't about bureaucracy. It's actually a tactic necessity, because you don't know which systems or comms or accounts they have access to or who has access to what. And beyond that, teams need different information.

Execs really only need to know or care about the business impact of an incident. Engineering teams need containment steps. And as your incident response progresses, you'll gradually, naturally tag people in. But you'll do this on a need to know basis, because not everyone needs to know the full details of your attack.

STEVE MCGHEE: Yeah. That's a really good way to put it, because I know I've heard the phrase need to know a lot. And it sounds like corporate junk. Like, come on. I need to know a lot of stuff. I'm a curious person.

JORDAN GREENBERG: Or like secret agent.

STEVE MCGHEE: Yeah, like.

JORDAN GREENBERG: Yes.

JESSICA THEODAT: All the yellow tape. All the red tape.

STEVE MCGHEE: Yeah. Yeah. But being aware of the idea that a reliability thing happens because a thing broke or a code got pushed in the wrong place or the config or whatever, that's a passive-ish system. The code's not listening to what you're talking about generally, I think. But in a security incident, that's always a possibility. It's possible that you have an active adversary in a security incident, which is fundamentally different. And that creeps me out. So I'm really glad that I'm on the reliability side and not the security side because that sounds hard. So hats off to you for actually going against the bad guys. I'd just like to say I'm a janitor. I just fix broken things. I don't want to deal with bad guys. That seems that seems hard.

JORDAN GREENBERG: It's being responsive and being more creative, as you and I have talked about. Jess, before. In a security incident, someone has gone about this in a way that's not follow the rules, It's change the rules, and break the rules to try to get that information. And reliability is about I set this up to work this specific way. I need to make this thing available and accessible, which is very counter to how a person who is attempting to compromise security thinks. It's a very different script versus no script to follow, I think.

JESSICA THEODAT: It's a good way to put it.

JORDAN GREENBERG: Yeah.

STEVE MCGHEE: Well, this has been a good conversation, but I think we need to start to wrap things up a little bit. So, I mean, how do we solve this overall? We've got two minutes to solve all of reliability and security. Like, go, this should be easy. What should we tell people who are listening? Remember, the listeners of the Prodcast are generally SREs who don't work at Google, or SRE adjacents. They're going to be DevOps teams, platform teams, software engineers, TPMs, CTOs, I don't know, lots of people. So what should they think about when they're comparing their reliability teams and their security teams are figuring out how to help them work together? What's the idea here, especially in the context of AI? Don't forget that. We forgot about the trend for a little while there.

JORDAN GREENBERG: Yes, we do have to add the trends.

JESSICA THEODAT: For sure, for sure. So I think what the takeaway here is to recognize that it's not a zero-sum game. Ultimately, we're working towards the same thing, ensuring that we earn and keep user trust.

STEVE MCGHEE: So if I'm a CIO or whatever it is, titles that people use in the world of VP or something like that, and I have a reliability team and a security team that I work with or work for me or something like that, who do I fund more? Who do I tell is in charge? This is a loaded question, obviously.

JORDAN GREENBERG: Yeah. That's a tough one.

STEVE MCGHEE: How do I get them to work together? How do I get them to understand that they're both important, or do I just say, good luck, everyone? Is there something here that we can give in terms of guidance around how these teams can work together?

JESSICA THEODAT: I think the guidance here is to understand the trade-offs between the different decisions and the optimizations that you're doing, and mapping those back to the higher business goals. Again, because ultimately, the reason why we're doing all of this is to serve a need, or to fulfill a need for our users. And those are usually best described as business objectives or goals. And so we want to make sure that the strategies and the direction that

we're heading in aligns to those things. And the way that you do that is by evaluating the different properties that speak to the goals and evaluating the trade offs between them, and understanding how each decision impacts users, how each decision may impact revenue or reputation, and so on and so forth. And then what you do is you prioritize for the most likely and most impactful failure modes.

JORDAN GREENBERG: Oh, and this makes sense, knowing that, oh, the thing that you expect to happen is the thing that you should definitely plan for. So it helps with the prioritization. It helps with understanding, oh, which thing do I have to invest in?

STEVE MCGHEE: Maybe we should do something about that.

JORDAN GREENBERG: Thank you so much, Jess, again for coming back to us on the Prodcast. We are so thankful to have you. And we're so thankful to have you fighting to keep our tools reliable and secure. Appreciate you joining us again today.

JESSICA THEODAT: Thank you. Thanks for having me back.

STEVE MCGHEE: Thanks for coming.

—

[JAVI BELTRAN, "TELEBOT"]

JORDAN GREENBERG: You've been listening to Prodcast, Google's podcast on site reliability engineering. Visit us on the web at SRE dot Google, where you can find papers, workshops, videos, and more about SRE.

This season's host is Steve McGhee, with contributions from Jordan Greenberg and Florian Rathgeber. The podcast is produced by Paul Guglielmino, Sunny Hsiao, and Salim Virji. The Prodcast theme is Telebot, by Javi Beltran. Special thanks to MP English and Jenn Petoff.