Scott Hanselman:

This is ACM Bytecast, a podcast series from the Association for Computing Machinery, the world's largest education and scientific computing society. We talk to researchers, practitioners, and innovators who are at the intersection of computing research and practice. They share their experiences, the lessons they've learned, and their own visions for the future of computing. I'm your host today, Scott Hanselman.

Hi, I'm Scott Hanselman. This is another episode of Hanselminutes, in association with the ACM Bytecast. And today I have the honor of speaking with Kelly Shortridge. She's a Chief Product Officer at Fastly. How's it going?

Kelly Shortridge:

It's going very well. It's a beautiful spring day here in New York.

Scott Hanselman:

It is a beautiful spring day. I've gotten some sunshine today and I feel a lot better. Everything sucks, but it just sucks slightly less when it's sunny.

Kelly Shortridge:

That is true and things are blooming. I can't complain.

Scott Hanselman:

Yeah, absolutely. So you are the author of *Security Chaos Engineering: Sustaining Resilience in Software and Systems*. And I spent the weekend reading the book and trying to understand where the intersection of chaos engineering and security engineering is, because I remember when Chaos Monkey was a thing and I just got to imagine all the Netflix people running around pulling cables and the monkey was just messing up their stuff. And now I'm trying to understand the intersection of security engineering with chaos engineering. I wonder if you could help me understand that.

Kelly Shortridge:

Yes. I think it's better characterized by the umbrella of resilience engineering, if anything. I've actually had the rare and delectable pleasure of unplugging cables from Fastly's POPs. Of course, network continued working perfectly. It is a thrill though, I will admit. Part of the title with the book with a little behind the scenes tea is chaos engineering, especially at the time, was a big buzzword. The book is certainly more than just chaos engineering. That is one tool in the resilience engineering toolkit. When we think about resilience, it's ultimately about how do you recover from failure of any kind and prepare for what's next? That what's next could be a threat, but equally it could be a business opportunity, it could be massive traffic growth for good reasons, or it's a DDoS. So security really is a subset of the both surprises, stressors, opportunities, and threats in a very broad sense that we need to think about.

Scott Hanselman:

This might be a dumb question. It could be a spicy question, but why call it security chaos engineering? Is it because those are fun words? Because resilience engineering wouldn't fly off the shelves? Because it seems very clear that resilience is really what we want, but it's just not a sexy term.

Kelly Shortridge:

I mean, I think this is a classic tension always when you're trying to publish a book or frankly a movie, you have to have some sort of catchy name.

Scott Hanselman:

No, that's a great point. Chaos would be an awesome movie name, but resilience is like, it's more of an A24 movie.

Kelly Shortridge:

Exactly. A24 vibe. It's also talked about at Davos and it's in the National Association of Corporate Directors book around organizational resilience. It is a great Latin root word for international appeal, but I think chaos, people are like, wait a second, chaos can be a good thing. That doesn't sound right.

Scott Hanselman:

Absolutely.

Kelly Shortridge:

Ergo the title.

Scott Hanselman:

I want to engineer chaos. That'll be very exciting.

Kelly Shortridge:

Exactly.

Scott Hanselman:

Now you have said that security should be designed for failure, not for prevention. And I think that's a really cool way to think about that. Can you think of an example where there's a perfectly secure system that still failed in the real world? What's an example where, oh, it still happened and we couldn't stop it?

Kelly Shortridge:

I mean, I feel like tons. First, no such thing as a perfectly secure system. I think there's so many esoteric failures out there. I think about the airline industry has learned many years ahead of software, about the intricate nature of complex systems and all the failures that could go wrong. But the example I always think about is the fact that they designed, I forget which airplane it was, which model. They designed it with safety in mind to almost every degree, except for the fact that in a very bizarre scenario, if you somehow exploded the coffee maker by boiling it too hot or something, it happened to be close enough to the panel with some cables that it could cause a critical failure while the plane was in air.

Scott Hanselman:

Oh, my God.

Kelly Shortridge:

Right. You wouldn't think about that as that is the trigger to a massive failure that means there has to be an emergency landing, but yet there they were. So I think looking at real world systems and all the just bizarre ways that they can fall apart. I remember back in the days with Twitter, there was Cyber Squirrel where it talked about all the power plant failures caused by squirrels just doing things squirrels do and how they were almost the more threatening, advanced, persistent threat because of the damage that they brought. I think there's just so many examples of your best intentions. Reality is stranger than fiction. You're not going to be able to dream up every scenario that's possible. So you have to prepare for the idea of, okay, things will go wrong. How do we minimize impact and make sure we can evolve to meet the moment?

Scott Hanselman:

Yeah. I think it's so important also to remember that, and maybe this is also a little spicy. It's on you to be responsible for your own resilience. And I remember in the early days of the cloud when we were all trying to get five-nines out of Azure and five-nines out of AWS, it's like, okay, Azure went down. I'm going to call somebody and yell at them. But how badly do you want your site to be up all the time? Do you want it badly enough that you're going to put it in both Azure and AWS? Do you want a copy of like how do you make a plane that doesn't crash? Do you fly two planes next to each other and then when one fails, like you jump to the other plane? It is all ultimately on us, is it not? And we just need to decide how hard to squeeze.

Kelly Shortridge:

I think there is usually a trade off if you want to really simplify it between cost and resilience. To your point, ultimately redundancy is multiple paths to get to the same goal. In practice, you need polyglot applications and systems that's pretty expensive to pull off. I do think though that software has a beautiful luxury we sometimes don't leverage. To your point about planes, sometimes you can run two instances of a service to like offload capacity in a way you just can't do with physical systems. Same thing with simulating failures too. Again, I think that's a very responsible thing to do. A lot of complex systems wish they could. For instance, you can actually instantiate a real kind of clone of the production system. You can't replicate a realistic clone of New York City to see if there's a certain level of trash blocking sewer drains, like what level of flooding will cause deaths.

You can't simulate that with any degree of ethics, but you can in the computer world, but we're not doing it. So I think that's part of my call to action that was big in the book is like, okay, how do we start taking this more seriously and really leveraging the benefits that the flexibility software begets gives us? So I do think there's, to your point, yes, some of it is more expensive to do, but in another sense, maybe we should be allocating more spend towards some of that simulation or just understanding the resilience contours of our systems better when other industries are just looking at us shaking as like, "Why aren't you doing this?" We wish we could do this. Yeah. I've been thinking about resilience in my own kind of personal IT life. I assume you have a home lab and various sorts.

Right now, as I talk to you, because I had an appointment with you, I am on my backup internet. Turns out I'm looking at my unify here. My WAN failed over at 4:38 AM and I have yet to diagnose it. So I'm on backup internet right now. And when I mention that to people like muggles, like regular people, they're like, "You have two internets at your house?" And I'm like, "This is my job, bro. I'm here." I've been doing this at this house for 18 years. It costs me 45 bucks for Comcast as backup internet. I have my fiber, but

the backup's for $45. It only has to fail once today and that made it worth the money for the year because otherwise I would've had to cancel on you and that wouldn't happen.

Exactly.

Scott Hanselman:

Yeah. So it was a choice. I feel like there are teams that think that they are resilient until the thing happens. What's the difference between a security team that thinks they're resilient and maybe one that actually is? I feel like there's a lot of false confidence in metrics theater that happens.

Kelly Shortridge:

Metrics theater, that could be an episode in itself. They're like, oh, what present security coverage do we have? Nobody knows what that means. It's a meaningless metric. That is a great question. I think the giveaway is when the security team feels a sense of control probably means that they don't have a lot of resilience because part of resilience is embracing the fact that there will be things well outside of your control. So it's like, how do you prepare for that? If you are trying to control everything and make things as deterministic as possible, you've already failed in my view because the world is not deterministic.

Humans aren't deterministic. We would like computers to be deterministic, but they aren't fully at least. It's one of the hardest problems in computer science is verifying that the software works the way that the designer of the program intended it to. So whenever I hear a security team say like, "Well, we have full control over the software delivery lifecycle." I'm like, "Are you sure?"

Scott Hanselman:

I just imagined a memed version of you and that one guy from HBO was like, "You sure about that?" Sure about that?

Kelly Shortridge:

"Yep. Kelly Shortridge says," Yeah, you're sorry. Exactly. Probably means you're investing in things that make you feel good and give you that sense of control and not the things that minimize impact.

Scott Hanselman:

Yep. It's not directly security, but that old joke of backups always succeed. It's restores that fail.

Kelly Shortridge:

Yes.

Scott Hanselman:

So it makes me think about chaos engineering and infrastructure is about pulling wires and yanking wires is very exciting, but I feel like there's a lot of pull the wire moments that can happen in security, but people are too scared to try.

Kelly Shortridge:

I mean, fear is pervasive in the culture and it's a disservice to the industry and the mission for sure. I think there are also cases where you could be starting with smaller experiments or just testing more basic hypotheses. My favorite leveraging actually Fastly's compute, which is like a high performance

serverless. You can think of it that way. It's just a little function that strips out cookies just to see like, hey, does your login site work? Same with offheaders.

It's just those basic assumptions you hold like, of course we're always going to require this for the login page. It's like, well, is that true? Are you sure? And especially when you can duplicate the request, which this prototype did, it's pretty low impact to the business to run that experiment. There are of course things where it's like, "Hey, RMRF, the customer database, yeah, that's going to be a pretty poorly designed experiment with high consequences, but there's such a range in between that I think it's very unfortunate that security practitioners are too hesitant to try those experiments, especially they're very hesitant to reach out to their peers across the island, like platform engineering and be like, Hey, can we co-conspire on developing some of these experiments?" Because there are a lot of jointly held assumptions too that aren't always poked and prodded.

Scott Hanselman:

You mentioned about determinism and how computers and software is not as deterministic as we would love to think that it is, not just because the software pretty much always runs as you wrote it, but whether or not your intent was well expressed certainly is a problem and then the environment within which it runs, you can't always count on. But I'm finding that people seem to be spackling or puttying over their systems now with what I'm calling ambiguity loops, which are basically using an LLM to deal with ambiguity by letting it fill the ambiguity with the randomness. I'm curious in your business when now people are running playbooks that aren't scripts, they are pros like a markdown file is not a script, I think you would agree. How do you feel about that? Is there a place for LLMs to live in security and in resilience and in chaos, or do they just increase chaos and entropy?

Kelly Shortridge:

It depends. I think they're only going to be as good as the corpus that went into them for one. And so unless you can verify really clean code went into it's like, well, it can maybe be a good basis for actually in some cases chaos experiments or specific configurations, integration tests, et cetera. What I will say though is to me the more important litmus test is this replacing human judgment? And if it is, that's probably not a good case for an LLM. I am pro human judgment and creativity. I am pretty anti-very repetitive work, very tedious work where you don't need that kind of judgment call. LLMs can be very helpful there. I think there are also document intelligence examples, who loves going through your compliance documents and pulling out relevant information like LMs can shine there and that way you can focus more on strategically like are we sustaining resilience?

What are the indicators we should be looking at for that? But asking LLM, is our system resilient probably is not going to be a great outcome. I think the markdown example is a little interesting because I am also pro making security more accessible. I think we dress it up in a lot of arcane key phrases and buzzwords when really a lot of people could benefit the industry and contribute. So maybe simplifying how they can enter, not requiring scripting knowledge could be a good thing. I also see it as a potential footgun. So I feel like I'm a little mixed.

Scott Hanselman:

No, I hear you. I'm with you 100% on the human judgment. It cannot be overstated. I assume that you're speaking to universities and early in career people often and you give your speeches and stuff and you talk to Spain like, "What should I learn?" It's like, "You should learn how to have good taste." How do you learn good taste? Well, you just got to get in there and get your hands dirty and do the thing and start pulling wires and figure out the system. Certainly I don't want to outsource things to human

judgment and toil, like keeping a site up is toil. SRE is toil, but SREs that are really good at their job are good at their job because of their judgment. So there's going to be this constant tension between the idea that you would replace an SRE with a markdown file makes me very nervous. But an SRE agent that could maybe kick the node and keep it running while I drive over there has value to me.

Kelly Shortridge:

Yes. Buying capacity and buying time I think is a great use case as well to your point. How do we help the human engage better with the system? Even just like the rubber duck problem solving, that can be a useful thing for the LLM. That does require quite a bit of expertise already though.

Scott Hanselman:

I love that you brought that up. I was talking to someone recently and I gave a whole talk and I brought up rubber duck debugging and like no one got it. And like, am I unc suddenly no one gets... This is not a generational thing. Talking to the duck. Yeah, your face is saying the same thing. Those are important moments to... You're talking to yourself in the mirror except now the mirror can talk back and that's really cool. I find that to be super helpful. Have you used LLMs in that context to figure your thoughts out and talk to yourself?

Kelly Shortridge:

Sometimes I use my cats more often for that because they give especially judgy looks that make you really question what you're throwing down. I think LMs can be helpful though, especially I see a lot of people struggle to get buy-in. This is kind of getting into corporate type stuff, but especially international companies where it's like, "Hey, I want to get buy-in on this resilience initiative. How is this going to resonate across different cultural contexts, for instance?"

Scott Hanselman:

Oh, that's a good one.

Kelly Shortridge:

Is chaos perceived negatively in certain nations versus others? I can tell you, for instance, when I talk about deception and using that as a technique for resilience engineering, security engineering, American security practitioners, not universally tend to result in like, "Well, we don't want to be the bad guys." Now the EU, they're like, "Tell me more. Yes, please. We want the subterfuge here." So that's kind of fascinating. And I feel like that's an interesting twist that I've found really useful with LLMs.

Scott Hanselman:

I like that. I didn't think about that. Yeah, you're right. It does see broader than we do and it can challenge your assumptions, especially if you tell it, challenge my assumptions as opposed to telling you that you're absolutely right.

ACM Bytecast is available on Apple Podcast, Google Podcast, Podbean, Spotify, Stitcher, and TuneIn. If you're enjoying this episode, please do subscribe and leave us a review on your favorite platform.

You probably work with big companies like banks and slower moving things, healthcare. They're a little more conservative. I'm curious, is there an example where traditional compliance actively makes systems less secure where they think that they're checking boxes but they're actually hurting themselves?

Kelly Shortridge:

Yes. Actually, a frequent co-conspirator of mine, Josiah Dykstra, wrote a paper, not with me, it's an excellent paper about that exact topic I think specifically covers HIPAA and maybe one of the others that shows that being more compliant doesn't actually result in better security outcomes. I'm very much of the view and I've tried to caution regulators as well as well-intentioned regulation in this space very quickly calcifies and ossifies it's what helped in year zero through maybe even year three may end up actually eroding resilience long-term.

Great example, I'll keep the person anonymous, very innovative CISO, had to explain I think over a few years to his auditors, actually it's a great thing that we don't allow SSH access anymore because that's what attackers love. They love when you leave the door open like that. But on the little compliance checklist for the auditors, they're like, "Okay, but it says you're required to have SSH access." And he's like, "Okay, but the security outcome is now better." So that's where sometimes it can actually hold companies back, even big companies who do want to innovate, but basically tying them to their investments and their spend to just checking those boxes, which is a disservice to their overall mission.

Scott Hanselman:

Yeah. I always want to assert assumptions. I feel like, because I work at Microsoft in my day job that my ignorance is kind of my superpower because someone will throw me into a new situation and I'll see a checkbox like must include SSH. But why? And no one knows. I don't know, 13 years ago, someone wrote that checklist and now it's a thing and then investors see it and compliance people see it and that checkbox is the thing that stands between you and some certificate or some badge and that's a problem.

Kelly Shortridge:

It's a huge problem and actually bring up a kind of elegant point. If you look at what resilience means across all sorts of complex systems, but also the ones we're talking about here, a lot of when a system is stuck in, let's say it's not elegant, but like a less resilient or unresilient state or fragile state is because a lot of the processes and practices that they have in place to your point are from an equal iridium that no longer exists. The status quo has moved on the practices haven't. And so, you're just continuing to erode resilience as you stick to this old world and have not adapted to the new one in the new context. The same with, I always hear CISOs being like, "Well, once we patch the vulnerability or fix it, then it's fine." It's like, well, if that actually resulted in an outage or a breach, it's not actually fine because you still haven't addressed the underlying impact.

You've just patched over the one way attackers got in. They haven't adapted to that new paradigm and that new equilibrium, which is hard. It's updating your mental model of the system, which is not easy. And that's why it is so important to have people who will be like, "Well, why? Why is that?" Just poke and prod.

Scott Hanselman:

Often CISOs and security teams make dashboards because they want to roll things up and the bigger the company, the bigger the dashboard and then the CISO has to really, they can't know the entire stack. The stack is now too deep. So what is an example of a misleading security metric or dashboard that might cause someone to make a mistake, they're relying on a dashboard, but it's maybe a misleading metric?

Kelly Shortridge:

So many metrics. Certainly that security coverage one or risk coverage. Also, the number of vulnerabilities discovered. I'm trying to remember who it was, who talked about this where actually when things started to get better, it meant that their application development teams were surfacing more security issues, which was a good thing. And it meant that there was more of that trust, mutual trust between teams, but it looked like it was getting worse.

Scott Hanselman:

Yeah. See, that's a great example. That's the whole thing like, "Oh, I get all these bugs and all these security issues this good stuff." All of that is low hanging fruit, but they'll assume that something bad has happened or something has changed and they're going to then correlation and causation are not the same.

Kelly Shortridge:

Exactly. And also a lot of the security specific metrics don't tell the bigger picture includes, I think about the poor platform engineering teams who are handed a list of a thousand vulnerabilities. Turns out a lot of them are in components that aren't even exposed to the public internet. Should they prioritize those? Probably not. And meanwhile, actually there are multiple cases of this, so I'll keep them all anonymous, but it's pricing actually how often this happens. Security team will be on them, fix all of these even if they're not publicly exposed. And then the security team actually maintains their creds into their whatever admin system or security system that has its hooks into everything is in a text file on their desktop.

It's like, well, what do you think is actually the bigger issue here in terms of what attackers could leverage? So there's a lot of that kind of attacker math, attacker calculus that isn't baked in as well. I think there's also, even if we think about business context, the metrics that a lot of security teams track and even CISOs track aren't the ones that the board wants to understand or other executives need to understand either.

Scott Hanselman:

Yeah. If I go to fastly.com and I click on products, you've got all the network services and all the things that Fastly is known for. There's a whole section on security and there's also, you have services and folks that you can hire professional services and things like that. But how should I think about what security is my responsibility and what is the responsibility of the vendor for whom I am paying a lot of money to make things secure?

Kelly Shortridge:

I think it depends on the vendor. I think in the case of, let's say some sort of SaaS application, let's take it sales and marketing so it's going to have some of your customer data, prospect data feels reasonable for the most part that the encryption and things like that should be handled by the vendor for sure.

There are cases I'll use like Fastly where we have the platform where you can basically write code, run code, et cetera. It's our responsibility and we have done this to layer in memory safety by design. Same with isolation models, ensuring safe multi-tenancy. It's very much our responsibility. Making sure that you don't write vulnerabilities into your own code. It's like, well, that's probably outside of our rebate, though that's where sometimes ProServe can come in. Things like, "Hey, you have spun up a service on Fastly and that connects to a database that you have wide open without any access control list."

Not really our responsibility because we don't touch that component. There are ways that we can help with middleware that runs on our platform. I do think though there's a fundamental principle though in the conversation a lot of people miss which is like you have to own your own dependencies and so what you adopt, you do have to basically think about it like, well, we have to assume at some point something will go wrong with it, whether that's a security issue or not. And I do see a lot of the hot potato game happening [inaudible 00:23:13].

Scott Hanselman:

Yeah. That's exactly why I asked you that question because I think that people pay a lot of money for a platform, a cloud platform because they want, as they say, a throat to choke. It's like, who gets yelled at? "Get Kelly on the phone. I want to know what's going on over there." But then it's of course their thing. Then they are bringing in who knows, unknown node packages from unknown provenance and then they haven't thought about their entire secure supply chain. But at the same time, I like your point about a sales and marketing CRM or something like that.

Is it their job as an app to be in charge of AI bot management or DDoS or even API security? That would be an example where a cloud platform could secure those endpoints and hide that. So I like the separation of concerns there, but I wonder if people who are putting together their own systems think about that. We shouldn't be in charge of API security. Let Fastly secure our endpoints and then they just have a nice clean, bright line or is it always layered and they would have to do that. Basically you have two layers.

Kelly Shortridge:

I think it really depends on the company and their level of resourcing. There are some companies that culturally want to own more things or build more of their things and so they'll leverage us more to be able to DIY. There's certainly others where it's like, well, let's just use what Fastly has. Especially when it comes to security, putting in, whether it's our WAF or like you said, the AI bot, like insights we're able to surface, have that in front of our services, apps, sites, whatever it is. It really depends on resourcing. There's also the element of, I'm going to say in newer worlds where I have seen so many security leaders thrust into a conversation now where their CEO, their CMO, their board is like, "Hey, what AI bots are actually trying to scrape our stuff so we can monetize it?" This thesis is like, "I've never had to think about this before." Trying to DIY that is pretty hard and hiring that expertise is some of the most expensive expertise out there right now using a tool probably makes sense in that case.

Scott Hanselman:

Yeah, I think that's a great point. I mean, this is the thing, what do we do here at the company? We do insurance. Okay, then why are we doing AI bot management? That's not our job.

Kelly Shortridge:

Exactly.

Scott Hanselman:

So I always think about the business and I think sometimes when we are talking about all the things that we've been talking about on this show, we don't talk about like, why did we actually make this software? We made it to solve X business problem. Therefore, what responsibility is mine and what can be outsourced by someone who actually knows what they're doing, whether it be Fastly or Azure or

AWS. Let somebody who actually cares about that, do that while I focus on the business problem. Honestly, I don't want to do this stuff Fastly does. That's why Fastly's good at it. You know what I mean?

Kelly Shortridge:

Yes. And it's laying out your own pop infrastructure, especially in this day and age of RAM prices being what they are, especially if you were a small business, it's quite unlikely that you're going to be able to do that, nor should you, because to your point, it's not your core business. I was thinking when you were talking about the example of the duplicate or secondary internet, part of that is because your essentially critical function hosting this podcast is to make sure you can record it.

Scott Hanselman:

Exactly.

Kelly Shortridge:

However, if you were to build your own microphone, I'd be a little bit like, "Is that actually your core value add here?"

Scott Hanselman:

No, that's a good example.

Kelly Shortridge:

It's just understanding what matters and what makes you unique as a business, for sure.

Scott Hanselman:

Yeah, absolutely. This is totally random and off-topic, but we had a phishing thing happen at work yesterday where they send us phishing emails, but it's from the red team and I was so proud of myself. I was just like, "I don't think that's real." And I was like, "Report phishing." And then it was like, "Congratulations. You're one of the better people. I don't know. There's some number of people at the company that does that."

And I'm always impressed that there's whole teams out there trying to attack us internally that I've never even met. You know what I mean? The red hats or the, I guess they call them blue hats at Microsoft because our badges are blue. Somehow I was just thinking about there's people trying to create chaos internally at the company and they tried to catch me yesterday with a phish and I did-

Kelly Shortridge:

However, what I will say, that has gone wrong in the past and I've spoken publicly before it was cool to have this take. People were quite angry. I remember this was many years ago where I said, "Hey, it's maybe not a great thing, especially during COVID, this happened a lot to be like, here's your surprise bonus plan and that's the phishing simulation email."

Scott Hanselman:

Oh no, that would be awful.

Kelly Shortridge:

Right, but that was happening.

Scott Hanselman:

That's punitive.

Kelly Shortridge:

I agree.

Scott Hanselman:

Click here for more money. No, this was not that. This was more like you have mail waiting for you in the mail room and I'm like, "We don't have a mail room."

Kelly Shortridge:

There you go.

Scott Hanselman:

That's fair. You're right. I mean, this is the whole sprinkling USB keys around the bank parking lot way of doing things. It's like Beyonce's new album, sprinkle, sprinkle, and then everyone plugs it in and then owns the entire bank.

Kelly Shortridge:

Yeah. Something like that. I think there are a lot of experiments. I think it's always keeping in mind again the human element that you don't want to sow distrust, but again, you can also make the experiments collaborative too, which that can get really fun because I've actually... You mentioned SREs. When I talk to real attackers, they're generally not scared of security engineering teams. They're scared of SREs because SREs will obsess performance. There's that one back door, was it XZ Utils where it was a guy who's like, "Oh, performance degraded by, I think it was less than 1%, what is going on?" discovering the back door.

Scott Hanselman:

Right. That was awesome.

Kelly Shortridge:

Work with them.

Scott Hanselman:

That was pretty cool.

Kelly Shortridge:

So I think security teams need to embrace, hey, you will have good ideas, but there are going to be other very clever people where if you say, "Okay, if you got really mad at the company, how would you attack

us?" They're probably going to have some interesting ideas that maybe can become experiments or clue you into some gaps maybe you have in your current security investments.

Scott Hanselman:

Very cool. You've given me a lot to think about. I thought that chaos engineering and security chaos engineering and this kind of resilience was a branding exercise, but it feels more concrete after having chatted with you.

Kelly Shortridge:

Yes. I mean, again, it was mostly a buzzword and it's part of playing the game that publishers have to play. I will say though for a very long time since I was the we lad, as they say, I've been obsessed with chaos theory and I do think chaos theory, which is quite beautiful in the sense of systems do have an order to them, but it's not necessarily predictable. It's more like obviously like a fractal or dragon curve in many cases as any meteorologist knows well. So we need to focus less on, again, do we have control over it? Are we able to predict it? The quote I love is from Susan Elizabeth Howe, who's a geologist who said, "A building doesn't care whether the earthquake was predicted or not, it either stays up or it doesn't."

Scott Hanselman:

That's good. That's very good. That's very good.

Kelly Shortridge:

I feel like that's the essence of it.

Scott Hanselman:

Yeah. I like that one. One of my favorites in a similar vein is a Babylon 5. The avalanche has begun. It's too late for the pebbles to vote.

Kelly Shortridge:

That is also very good. Yes.

Scott Hanselman:

I don't like this. This is not a good idea. This is happening. Sorry, this is happening. So buckle up.

Kelly Shortridge:

Yeah, exactly.

Scott Hanselman:

Well, thank you so much, Kelly Shortridge, for chatting with me today.

Kelly Shortridge:

Thank you for the great questions. Appreciate it.

Scott Hanselman:

We have been chatting with Kelly Shortridge, the chief product officer at Fastly. This has been another episode of Hanselman's in association with the ACM Bytecast and we'll see you again next week.

ACM ByteCast is a production of the Association for Computing Machineries Practitioner Board. To learn more about ACM and its activities, visit acm.org. For more information about this and other episodes, please do visit our website at learning.acm.org/bytecast. That's B-Y-T-E-C-A-S-T. Learning.accm.org/bytecast.