EPISODE 1114

[INTRODUCTION]

[00:00:00] JM: Logs are the source of truth. If a company is sufficiently instrumented, the logging data that's streams off of the internal infrastructure can be refined to tell a comprehensive story for what is changing across that infrastructure in real-time. This includes log-ins, permission changes and other events that could signal a potential security compromise.

Datadog is a company that was built around log management, metric storage and distributed tracing. More recently, they've also built tools for monitoring the security of an organization. Detecting security threats can be achieved by alerting or known security risks or pieces of information that could be indicative of a vulnerability.

Mark Tremsal is an engineer at Datadog who joins the show to talk through security monitoring. Full disclosure, Datadog is a sponsor of Software Engineering Daily.

[SPONSOR MESSAGE]

[00:00:53] JM: Thank you to Datadog for sponsoring this episode. Datadog is a monitoring, security and analytics platform for developers, IT operations teams, security engineers and business users. With recent security monitoring product developments, Datadog now provides one integrated platform for all your monitoring needs and it includes security. Security monitoring product allows you to easily detect malicious activity in real-time before it affects your customers. Teams can investigate security threats using OOTB detection rules and detailed observability data.

Metrics, traces, logs and more in one unified platform, and you can learn by more by signing up for a live demo with one of their security engineers and you will receive a free Datadog T-shirt. Just go to softwareengineeringdaily.com/datadogsecurity. That's softwareengineeringdaily.com/datadogsecurity.

[INTERVIEW]

[00:01:54] JM: Mark, welcome to the show.

[00:01:56] MT: Thanks for having me, Jeff. Pleasure being here.

[00:01:58] JM: We're talking about security today, and I'd like to start off with a broad question. What does security mean in a world where everybody is on the cloud?

[00:02:07] MT: That's a great question. Historically, the security team was looking at the entire stack and had full responsibility for essentially everything. Nowadays, when you think of building on the cloud for the cloud services you use, this is a shared responsibility. So something we hear a lot about is not really, "Oh, I'm worried about AWS or GCP having vulnerabilities," but really my own developers and operations potentially misconfiguring one of these services. And so then an AMI is open to the Internet and all things like that, right? o a lot more about configuration [inaudible 00:02:50] that can lead to data leaks, for example. And at least at the cloud level, a bit less about purely the security of the services themselves, if that makes sense.

[00:03:03] JM: Interesting. So, the un-relying systems, you can generally trust them, but you may not be able to trust yourself to configure it properly to be secure.

[00:03:12] MT: Sure. And think about it, Google Security Team is best in class, massive. You would be [inaudible 00:03:20] to reproduce this kind of expertise for your own stack running in your own data center. But if you don't understand identity and access management model of GCP, maybe you make a misconfiguration mistake and GCP is going to have a hard time protecting you against that, right? So it's a lot more on these configuration issues and a bit less about actual techniques that attackers can use against you, at least for that level. Everything you build on top obviously very much the same as before.

[00:03:52] JM: What are the kinds of security threats from an external perspective that companies, organizations should be worried about?

[00:04:03] MT: I think a lot of these techniques, they haven't necessarily changed as much. But the way they get implemented is probably different. What would be a good example? Let's say

somehow you got a foothold in the environment as an attacker and you're looking for valuable data to exfiltrate. Usually you would check – I don't know, what NFS you have access to. But here, you're in the cloud. So maybe you'll enumerate S3 buckets instead, right? And so if you see an EC2 instance enumerating S3 buckets, maybe that's a scanner that you have for compliance reasons. Maybe that's something else. That's an example of a specific technique that you would be able to detect. It's interesting how the objectives and the techniques somehow knew, but by enlarge, a lot of the same ideas, a lot of the same goals that attackers could have still exist. But now to be able to detect them, you need a pretty good knowledge of how each cloud provider works, if that makes sense.

[00:05:07] JM: How did the vulnerabilities on the application side, maybe like the mobile application, differ from the infrastructure side, the backend side?

[00:05:16] MT: That idea is still very much on responsibility as a developer, right? There, you're a bit less worried about configuration [inaudible 00:05:25]. And more about not just vulnerabilities, but also increasingly you see security teams worrying about overall forms of abuse. And how can I protect my own customers, for example? Of course, you still have like the OWASP top 10 and a SQL injection and things like that, and how could an attacker get in.

But increasingly at the application layer, you see teams, security teams, I mean, worry about things like account takeovers. So one of my customer reuse his passwords. No one should, but that happens a lot in real-time. And suddenly you see a specific IP or a set of IPs trying out username password combinations across your environment. And most of them fail. But if a handful succeed, maybe that's a successful credential stuffing attack, right? And that means potentially someone now has taken over the account of one of your customers. Is that your responsibility to protect your customer against this kind of attack? Ideally, yes. And this is fairly new. This is not necessarily something that purely application security was looking at before. But increasingly, when we talk about application security, we talk about this form of abuse as well.

[00:06:42] JM: What are the role of the cloud providers in modern security? What guarantees do the cloud providers have to give to the application developers?

[00:06:53] MT: Usually they're pretty explicit about that. They'll have a shared responsibility model where they say specifically this is what I provide you in terms of guarantees. Actually a lot of that is backed by certifications and [inaudible 00:07:07] compliance frameworks and so on. And then they'll tell you, "To achieve these objectives, here are good practices that you should implement and things you should pay attention to."

It's interesting, right? You still need to trust that obviously the services are safe to operate. But we're talking about massive, very well-funded security teams with highly scaled people. Really, it comes down to do I understand what this service does. And the security model for it, right? Again, identity and access management in particular is quite hard to understand and very different from cloud provider to cloud provider. And it's pretty easy to make a mistake and not realize that through a combination of parameters, it turns out this S3 bucket is actually accessible from the internet.

[00:08:01] JM: And in this situation where I am running my infrastructure on the cloud provider, I make some mistake in configuring my security settings. What kinds of things can go wrong and how do I ameliorate those problems?

[00:08:20] MT: More and more, you'll see security providers actually try to be proactive and tell you. I think a good example is, last December, AWS started this service called IAM Access Analyzer. And the goal of the feature is to tell you if some resources in your environment are wide open to the internet or other AWS accounts. I mean, the reason for this service to even exist is because that's something that happens a lot, and by accident, and they used to be the cause of not just concern, but actual incidents quite a bit.

The way they approach the problem is a separate conversation. It's quite fascinating. Using formal reasoning, and I think a lot of the researcher [inaudible 00:09:05]. But the key point there is they are now delivering services that will try to actively tell you if the way in which you use all those services is potentially dangerous and not what you intended.

Of course, it still remains your responsibility to read the documentation and be careful. Or by supporting services that will help you put guardrails in place. But more and more, the cloud

providers themselves, they have security services that will tell you about potential threats that exist in your environment and even potential misconfigurations, at least the more common ones.

[00:09:47] JM: What's your background? How did you get into security monitoring?

[00:09:51] MT: I'm part of the product team at Datadog. We're a SaaS company. We provide engineers with monitoring and analytics so that they can keep applications up and running and secure. Before that, I was actually part of the team looking at how to secure the application itself. Before that, I was a consultant. But very much the concerns of the security teams in speaking with them and looking at what Datadog was doing at the time mainly for developers and operations around detailed observability data and how that could be used by developers and operations. That also started a number of conversation around using that same data to detect potential threats and investigate security issues. And more and more, we had customer demand for Datadog to also address the use cases of the security teams. So all of engineering, not just developers and operations, but also security teams as well. And that's the inception of what is now the security monitoring product, which is what I'm focusing on right now.

[00:11:02] JM: Datadog historically has done lots of monitoring, logging and APM. So what additional steps do you have to take to implement a security monitoring system?

[00:11:17] MT: That's a good question. At its most basic level, a lot of the data is directly useful, which is very interesting, because traditionally a lot of security products were very focused on logs and events, right? Consolidating them, normalizing them, analyzing them, and that was the foundation for a lot of security. Modern observability has a few different tools that can be used for different use cases. You mentioned APM. So you have traces and you're going to look at a single transaction from end to end across a distributed environment, and that's a great way to look at causality in the environment.

For operational issues, for example, some latency increase as perceived by the end user and maybe it's caused by some unoptimized SQL query all the way back towards the database. And it's very interesting to see how a lot of that detailed data and these new tools that you have can also be used for securities. Logs and events remain the foundation of a lot of that.

The things that conceptually – At a very conceptual level, what's different is that operational issues, you tend to look for fairly noticeable trends in the sense that even if you're talking about something very localized like an increase in latency for a specific browser version in a specific country, [inaudible 00:12:46] the intersection of a number of factors contribute to an operational issue. You're still looking for something that's very organic, so to speak.

On the security side, you are working against an intelligent attacker, at least very often. A lot of it is automated. But you're looking against people who are going to try to hide their tracks and be deceptive. So the kinds of technology and algorithm you use, it's a lot more about finding the needle in the haystack whilst you're keeping a single to noise ratio very good and a bit less about finding these major patterns that try to explain more significant changes in the environment. Do you see what I mean?

[00:13:30] JM: I do. And tell me more about how security teams would be using a product like this.

[00:13:40] MT: There are a couple of things here. The first one is we say security teams, but really security nowadays for most companies we see, it's also the responsibility of developers and operations. It's shared responsibility or shared burden. And so there are a few things there. One is security teams, they're not just using the product, but also trying to enable the rest of the organization. If you think of – I mean, for us, on the engineering side, it's easy to look back at the DevOps moment. And now in hindsight, a lot of these changes were very obvious. But you think about what used to be sys admins and figuring out how to apply engineering mindset and help developers deploy and run software reliably.

Well, what's happening right now with security teams is very similar. You no longer really have these teams that just say, "No. Don't deploy this. It has vulnerabilities X and Y that have been found. We're going to do a pen test. It's going to take weeks." They're more focused on how can I bake a lot of my security expertise into the tools that developers and operations already use so that they have visibility into what they're doing. As they iterate on software versions and so on, they can also use these iterations to other specific vulnerabilities that are surfaced or implement some good practices that I have identified. It's really more about enabling.

Now, a lot of the more traditional responsibilities of the security team remain. They're still by and large the ones writing the logic for threat detection, for example, for enforcement of configurations that are known to be safe, also for responding to potential security incidents. But to go back to your question, how do they use this data and these tools? The long answer that I just gave, it really boils down to enabling developers and operations to really do their job and have security be a part of that, and then also use that more detailed observability data for their more traditional responsibilities.

[SPONSOR MESSAGE]

[00:15:58] JM: Operations teams can find themselves choosing between two options, either take full control over the infrastructure yourself or give all developers permission to access production. The first approach increases the operations teams workload, which often results in overwhelming situations and ops becoming a bottleneck. And the second approach allows for rapid deployment of changes to production, but it causes a serious risk to infrastructure uptime.

With Octopus Deploy, operations teams have a third option. The Octopus platform can be authorized to run approved steps and play an intermediary role. Octopus delivers self-service without sacrificing control over production, and it also provides a comprehensive audit log of the changes that are being made. Developers can enable self-service with automation. By automating the processes that are forming a bottleneck, developers can free themselves from the waiting game.

You can learn more about run book automation at octopus.com/runbooks. Octopus can help you with your run book automation and just go to octopus.com/runbooks to learn how.

[INTERVIEW CONTINUED]

[00:17:08] JM: I saw an interview where you said that people are moving into a world in which security must become a shared responsibility across engineering teams to address the increased complexity and scale of environments. What do you mean by this idea of shared responsibility?

[00:17:27] MT: Well, going back to this example of how the DevOps movement change the relationship between developers and operations, and it's very much the same thing here. You had developers pick up pagers, and now they're passionately responsible for making sure software is reliable. And that's how you align incentives between developers and operations, right? Because I'm no longer trusting software to sys admins, to ops and hoping that it runs. If it doesn't [inaudible 00:17:52] problem, I might get paged for this. So I have skin in the game. On the security side, it's the same thing happening. And the reason for it, I mean, it's really the complexity and velocity of the environment. It's just changing so fast.

If you have a security team of a fixed size and you have ten times as many engineers in your organization, 20 times as many engineers, and now they're walking on systems that are very complex, especially with microservices and containers, and they're deploying multiple times a day. I mean, there's really no chance that I'm going to be able to stay on top of all of the things that are happening. I'm going to have to equip the people who know their systems best to be on the frontline of that and have some ownership of the security process.

So, we're seeing a lot of these transformations that took maybe 10 years for the DevOps movement to happen being very accelerated right now for security teams where maybe two years ago we had some of these exchanges with customers and companies already. And now it's even more conservative enterprise security teams. As soon as they migrate to the cloud, I would say, they're starting to look at security as this shared responsibility and how can I, with a size of a fixed – With a team of a fixed size, how can I enable the rest of the organization. Instead of doing, how can I automate a lot of that knowledge in terms of guardrails, in terms of this detection, in terms of remediation as well?

[00:19:32] JM: The term security monitoring, I'd like to go a little deeper on that. Security monitoring works on detecting whether logs match certain detection rules. Yeah, I just like to know more about how that works and how it's useful.

[00:19:50] MT: We try to have product names that are pretty descriptive, and the product very much monitors the security of applications and environments. That's the most straightforward we could find to describe that. The rest of the market, you'll sometimes hear security analytics or a combination of all the words. There's certainly like a larger market especially on-prem

called security incident management, or SIM, that also has quite a bit of overlap with some of these use cases.

How to do that? Yeah, at the most basic level, you are looking at the environment. You're looking at the environment in two main ways. One is changes that's all happening in the environment, and lots are really a good way to convey that. The other one is because looking at changes obviously only catches what's moving, especially for compliance, the other way in which you look at the environment is to look at the entire state of the environment in a continuous manner and whether some aspect of that state could be noncompliant for external frameworks, but also potentially unsafe internally.

Analyzing that, well, you have a number of techniques that you can use to analyze this data and the events of the environment. Certainly, a foundation f that is to look at events collectively for a specific stream, let's say all authentication events and try to look for specific patterns. For authentication events, maybe you start to look at bare user spikes in fair log-ins. Very high anomalies could be a brute force attack. If that ends up with a successful authentication attempt, potentially the attack [inaudible 00:21:37], then now you have an account takeover to deal with. On the engineering side, at the most foundation level, you're really talking about asking stateful queries over a stream of data, specially a stream of ingested events.

[00:21:56] JM: Collecting logs and metrics and detecting whether they fit to certain rules in real-time has some engineering difficulties. Can you tell me about the engineering challenges for implementing this?

[00:22:10] MT: Yeah, absolutely. First of all, there's the log management layer of that. How do you safely intake data, process it at very high scale, very low latency? Make it searchable? And that's the more general observability world. I mean, that's really one of the main challenges that observability products try to solve, is all of that. Being able to do alerting on top as well and analytics.

From a security standpoint, the kind of analysis where you look at a stream of data to run stateful queries, this is sometimes called complex event processing. In the open source world, you might look at something like Apache Flink, for example, as a good example of the ability to

essentially ask SQL queries against that stream of data. How do you do that? Now it's not like batch-based. It's really as the data flows in and you start looking for those patterns. These are still fairly modern technologies. I mean, we don't use Flink per se, but the concepts apply.

They are not easy to develop. They are not easy to maintain, but to answer some of the challenges around real-time detection, very high cardinality in terms of entities. I'd say it's one of the technologies that can help answer these challenges.

[00:23:39] JM: Have you spent much time getting into the implementation of that? The Flink or other stream processing systems that are used?

[00:23:46] MT: Yeah, absolutely. A big part of the security monitoring product is an implementation that's security-focused, but for this specific kind of logic. And I say security-focused, but it's a platform approach and there are operational use cases that look very similar to that as well, although without the pattern matching or anomaly detection aspect of that, right? If you think of, for example, calculate – We're essentially talking about a distributed, like MapReduce operation on a stream.

For example, can I take all of the events that are part of a single session and continuously build that station as an object that is going to be viewable and searchable within the application? And this is not a security example. This is another – [inaudible 00:24:37] would be another use case that would be quite adequate for this kind of complex event processing technology.

[00:24:46] JM: And with all these logs, you have to decide how to index them in order to sort them. How do you index and sort logs that are coming in so that you can scan them properly for security questions? Detect security detection rules?

[00:25:02] MT: Datadog is a big unique there. We don't actually make this determination on behalf of customers. That's something, the product, an architecture we call logging without limits. And what we do is we actually decouple ingestion from indexing. So we're going to ingest all of the logs. We're going to parse them. Normalize them. Enrich them as well with GUIP, lookups, threat intelligent and so on. And that data that has been enriched, we're going to send all of that to customer archives. We're going to let customers search that in real-time.

Essentially a tail –f across your entire environment. Now, from a security standpoint, with the security monitoring product, we're going to make sure that the threat detection rules actually apply to the entire stream of ingested logs after they've been normalized and enriched.

Here, there's no choice to make for us of all the customer. We'll just normalize, enrich and analyze all of it. Now, after that, customers do decide what portion of that is variable to them at a specific point in time and they'll write features and say, "Okay. I don't really care about debugged logs most of the time. Please index everything else. And if they change their mind, that's fine. They don't need to redeploy anything. We actually make this determination or we let them decide what to index directly in the app. So they might decide to say, "Oh, okay. It looks like it has an incident going on. I want to see the shape of these debug logs that I won't usually care about. I'm going to index 1% of them just to see samples of what's happening and whether there's a trend over time. Yeah, the short answer is we don't actually decide for customers. We empower them to make the determination. As far as I can tell, that's pretty unique in the market. But it's one of the features that our customers seem most excited about. I mean, this control over cost and over what's variable to them and how to maximize value for them.

[00:27:07] JM: Besides the real-time detection of rules that might appear in logs such as problematic log-ins or problematic files entering certain domains that might cause a security risk, what are the other kinds of requirements that you need to have for a security monitoring system?

[00:27:30] MT: There's a lot. I think it's useful here to take a step back and look at some of the goals that the security team might have starting from simple visibility. I want to know what's happening in the environment, then to detection. Let me know if something malicious or anomalous happens. And this is where the pattern matching technology we talked about makes a lot of sense. Beyond that, you have things like investigation and response and building resiliency. Sort of a maturity curve from visibility, to detection, to investigation and so on.

Accelerating triage and investigation, there's a number of ways to look at that. But a lot of it comes down to making it very, very easy for users to pivot on underlying entities, right? You see that there's some kind of signal, an EC2 instance has a CPU spike and it's an outlier compared to the rest of the cluster, which is interesting. You can't quite explain it. Hey, is anything else

happening on this EC2 instance? And maybe you have a finding from a [inaudible 00:28:38] security product? Let's say GuardDuty and it's telling you, "Oh! Actually, this instance is calling out to a domain known to be attached to a crypto mining scheme, right? Okay. That's why I have a CPU spike."

That's like very simple example. But the ability to pivot on entities – By entities, I mean anything that's related to the attacker, like IP, the ASN, the user agent, but also the targets, right? So, host containers, accounts, and being able to use these map of entities to see, to visualize essentially the graph of security signals and how they could be related together to retrace an end-to-end attack. That's a big part of how we can accelerate investigation.

A very simple inclination of that, by the way, is just a dashboard, right? You have an IP that's involved in a signal and you say, "Okay, just tell me everything you know about this IP. And now you're looking at all of their logs tied to that, but you're also looking at potentially flow logs. You're looking at the list of simple transactions that were captured where this IP was involved, whether there's any threat intelligence tied to it all in one place. Even something as simple as a dashboard with all of that information in one place, that's already pre-aggregated for you. That's where a lot of team start when they think about accelerating investigations.

[00:30:06] JM: The landscape of products out there, there are a lot of different tool for security, security monitoring. This is more of a vertically integrated security monitoring and logging solution that comes with the other Datadog tooling. Can you explain how the vertical integration compares to how other platforms might work in case I'm piecing my own platform together? And there are advantages to having control over that total configurability?

[00:30:45] MT: Definitely. So if you're piercing your own security stack together, you're probably doing two main things. One is you're building some kind of real-time detection and alerting pipeline, essentially an ETL pipeline. There are open source products that help do that for, by the way, things like stream alert, which I think was invented over Airbnb. And that's usually coupled with some kind of way to ask ad hoc queries over large volumes of data, especially in the past, essentially your data lake. If you want a managed offering for that, in AWS, maybe that's Athena. On GCP, maybe that's something like BigQuery. We definitely see some teams try to roll their own stack.

There are a couple limitations with that approach. I mean it works really well if you have the expertise and the time. But the main one is that security teams usually would rather focus on something else rather than building out the platform from scratch. They would rather focus on actually writing the detection logic and running the investigation and building guardrails. Certainly, having something that's more turnkey goes the wrong way.

But the other thing there, it's a lot of data, and a lot of that is already being collected by developers and operations. So, duplicating it, there are cost and performance concerns associated with that. Finally, again, who's going to consume that data? Well, if you think it's going to be the security team and just the security team, then having a siloed stack makes a lot of sense. Certainly, that was the case for many, many years. But if what's you're looking to do is expose that data to developers and operations, you need to find them where they work, right? It turns out that's already where a lot of the data is. So then why duplicate it?

That's why at least for us, it was very, very obvious that there was this need in the market and we had security teams and also developers and operations asking about us tackling their security use cases. It's because of this consolidation. You could say there's some amount of normalization of security within the larger engineering organization. I mean, at Datadog, the security team, right? We're trying to solve security problems and challenges by applying engineering techniques. If that's how you think about security, well, then becomes harder to think, "Oh, okay. I'm going to have completely siloed tools and practices." And this is where this, you said, vertically integrated approach. And I think that I agree with that. This is where this kind of approach makes the most sense.

[00:33:28] JM: The volume of logs that can be collected and scanned in this kind of situation can be voluminous. Do you know much about the scalability properties of how you scale the log management systems along with the security monitoring situation?

[00:33:48] MT: That's the secret sauce, isn't it? It's a health problem. It's a big part of what we do, and it's a big part of why customers come to us. That's very much our expertise. I'll say this. It's easier to sale these systems as a SaaS product where you have entire teams [inaudible 00:34:05] to the problem across aggregated volumes to some extent.

If you're part of a team within a specific organization and there's only one or two people running these systems and they become critical, and now you're on-call and there are two people. This is where the DIY approach breaks down a little bit, is when the systems become very critical. Certainly, observability and security systems are considered business-critical nowadays.

So how we do it, I can't really get into any of the details, but the short answer is that it's just easier to do that as your main business with hundreds of engineers trying to solve that problem than it is with a much smaller team that's focused on shipping features, and observability is just a side job.

One last thing is, for us, when we think about scalability for security. I mean, if you can scale to handle all of the observability data that you generate for DevOps use cases, security doesn't necessarily generate more information on the contrary. In many ways, when we approach the design of the security monitoring product, outside of the complex event processing part, a lot of the scalability challenges were very much solved on the more fundamental level of intaking data, processing it, indexing it, archiving it. It's an integrated platform, and that made it much easier for us to focus on what was unique about security as supposed to the scalability and reliability challenges, which are shared across all observability products and use cases.

[SPONSOR MESSAGE]

[00:36:01] JM: Today's show is sponsored by StrongDM. Managing your remote team as they work from home can be difficult. You might be managing a gazillion SSH keys and database passwords and Kubernetes certs. So meet StrongDM. Manage and audit access to servers, databases and Kubernetes clusters no matter where your employees are. With StrongDM, you can easily extend your identity provider to manage infrastructure access. Automate onboarding, off-boarding and moving people within roles. These are annoying problems. You can grant temporary access that automatically expires to your on-call teams. Admins get full auditability into anything anyone does. When they connect, what queries they run, what commands are typed? It's full visibility into everything. For SSH and RDP and Kubernetes, that means video replays. For databases, it's a single unified query log across all database management systems. StrongDM is used by companies like Hurst, Peloton, Betterment, Greenhouse and SoFi to

manage access. It's more control and less hassle. StrongDM allows you to manage and audit remote access to infrastructure. Start your free 14-day trial today at strongdm.com/sedaily. That's strongdm.com/sedaily to start your free 14-day trial.

[INTERVIEW CONTINUED]

[00:37:29] JM: So if I'm a user and I want to implement the right rule set, the right things to scan for as I'm setting up security monitoring, how do I determine what to do? What to set up and what to be looking for? What kinds of threats are we looking for?

[00:37:50] MT: Outside of the product, there are number of communities that share their knowledge and even package these techniques. For security specifically, there's an open source format that we see now and then called Sigma that's going to try to convert these rules across different products and technologies. Within Datadog specifically, I mean, detection rules are just called, right? Essentially, each one is a JSON file. So you would manage it the same way you would everything else, which is as code. We have APIs for them. Down the road, probably it's a [inaudible 00:38:27] provider. So if you are managing the versions of your rules in your repository, you already have a change management process for it. Again, it's just a normal engineering way to look at iterating on this kind of logic. For techniques that are widespread and that we know you're going to care about, we'll also just package that content on behalf of customers so that they don't have to reinvent the wheel.

There's a fairly popular framework that security teams have been using recently called the MITRE ATT&CK framework, that is going to look at all of these potential techniques that attackers might be using. And when in the lifecycle of an attack they're using them, what they would call tactics. All the way from intrusion, to natural movement, to exfiltration of data.

For each of these techniques, there are ways to detect them depending on the underlying technology. Certainly, if we know how to detect that, we're just going to package that knowledge that it's there when you start using the product. But, yeah, there are a number of communities out there that share this knowledge. It's publicly outside of the research that's specific to what we do. Some of the basic, if you're just getting started and you're a smaller organization, there's a lot of knowledge shared on the internet by people who research these things very actively.

[00:39:55] JM: If I'm a user and I'm detecting a threat, what is the way to respond to that? Maybe describe a typical threat and what my response would be as a developer working on this system.

[00:40:10] MT: That's a great question. We sometimes call that step where you first see an alert or a signal about something that might had gone wrong. We sometimes call that triage. And there are a couple of determinations that you can make of actions that you would take. The first one obviously is if something is obviously malicious or a compliance issue, you're going to start investigating it and treating it as an incident. This is very similar to the experience of being on-call on the SRE side or on the developer side.

If it's not obvious what's happening, you're starting an investigation. Everything you do might eventually turn into an incident. You just don't know maybe the impact yet or the severity. So that's one branch of that decision tree. It's also very possible that you were alerted and something turns out to be benign. Maybe for example it's non-good behavior. Yes. These instances are doing something weird, but that's because they're part of my CICD pipeline and I know they have a compiler that's actually completely normal. What you're going to do then is you're going to say, "Okay. For all of these rules that detect these specific techniques, I'm going to allow entities that have these specific tags, like the CICD environment, to not trigger or to mute the resulting alerts and signals."

Another potential determination that you might make is that you have a false-positive. Meaning, the rule detected something that you were not actively trying to detect. The answer there is usually to flag the data, look at the underlying data and use that as a test case to write an updated version of the rule that hopefully is going to not have this issue in the future. Meaning, it's going to have better signal to noise.

Finally, the last case, you don't typically do that as a triage activity. But if you have a false negative, if you actually miss something. I mean, you're mostly likely going to write a new rule. That's not something you would find in triage usually, and this is part of a larger incident. Maybe you're threat hunting, right? You believe you have detection for specific kinds of techniques and it turns out that your red team or an external pen tester actually tried that and you didn't see it

well. What kind of data is available to me to write corresponding logic? Yeah, triage. Think of it as like the 30, 60 seconds it takes me to review a specific alert. And these are some of the branches in that decision three that I can make.

[00:42:47] JM: What's the retention period for security data? How does it differ from retention periods for other logs and metrics?

[00:42:55] MT: Obviously, that's up to you. At least for us, we retain all security signals for 15 months as a baseline. And the underlying data, especially logs, typically customers will want – It depends, like 30, maybe 90 days of warm data for things that are almost always useful. And then they'll want 1 to even 7 years depending on external compliance frameworks that they apply of archives available for forensics and compliance.

But even for the signals themselves, the reason that you typically have a fairly long retention, in this case, like 15 months for signals, is because I think we don't necessarily on the developer and operation side, we're very focused on real-time data. It's a big more unusual to go back a year. The only example is you're preparing for Black Friday and you want to see what kind of spike you have the year before. And so you have metrics for the best 15 months, and that's great, because it helps you do forecasting from year-to-year, right? But that's mostly an exception. The rest of the time, you're focused on real-time data.

On the security side, interestingly, the average dwell time, like the average time for an attack to be detected from the moment it started and the attacker got a foothold in the environment to the moment the company is actually aware of it, it's something like 200 days, right? A lot of these attacks take a long, long time for the company to be notified by a third-part or for the attacker to trigger an alarm maybe they try to exfiltrate data. Something fairly late in the lifecycle of the attack. Having long-term data there is very useful to be able to connect the dots especially if we're talking about one of these slow attacks that unfolds over a number of places in the environment and the attacker pivoting from place to place all the way into a moment when they find something variable. Yeah, definitely a different timeframe, a different time scale than for operational use cases where real-time data is the name of the game.

[00:45:05] JM: What are the outstanding issues with the platform? What are the sources of technical debt or what kinds of problems are you focused on today?

[00:45:16] MT: That's a great question. I mean, obviously there's no shortage of things to do on any product. We have a request for so many things. But there are a few classes of problems that I think are interesting. One is we talked about looking at patterns for potentially malicious activity, right? And on the security side, there are more advanced form of detections that you can use, things like anomaly detection or even machine learning. I know the term is used left and right. But in this case, I mean, actual machine learning where detecting a specific technique maybe at the network layer, it makes more sense if you're looking across a number of dimensions, like the size of packet and the number of domains to connect to and to detect something like someone beaconing over DNS to connect to a [inaudible 00:46:15] data, let's say.

The really interesting thing there is that some of the data science algorithms look a bit different for security use cases. Again, you're looking for the needle in the haystack. Not necessarily for the more noticeable patterns that are relevant. So, for example, if you think of anomaly detection, you're going to end up having very high community data that can be quite sparse, right? There are certainly domains there where there's a lot of interesting R&D around tweaking algorithms so that they make sense these use cases and not just for operational ones. That's one example.

I don't know what another good example would be, but we talked about connecting the dots based on the underlying graph of entities to reconstruct entire attacks that go into end-to-end. That's another domain. Assisting and automating investigation, that's definitely another very interesting challenge. And the reason for that is that you are looking to correlate data not just on fixed data points. But yeah, the underlying set of entities is essentially a graph. So it's a different data model, and how do you wrangle with that efficiently at scale to make sure it's still very fast and very relevant? That's certainly something that occupies a number of very smart people on the team. I'm on the product side. So I just have to say this is what we should be doing and the show is left as a question to the reader very often, or in this case, an engineer.

[00:47:55] JM: Well, as we close off, is there anything around security monitoring that you think will change in the future? Any subjects that you'd like to bring up as we come to a close?

[00:48:06] MT: I don't necessarily think things will change fundamentally. I mean, if anything, the product is a solid foundation for what it does. But the practices around security, that's changing very fast. For the past few years, the focus around baking security into other engineering practices, that has been very focused on the CICD pipeline. It's a natural point, right? Security teams look at this and think, "Okay. Anything that goes into production, now I have control over it. I can look for vulnerabilities. I can run a number of checks. I can blog developers if something goes wrong.

And I think we're seeing a pretty big shift where you still need to do that, obviously, but there's a limit to how much you can claim your infrastructure is immutable, when you start taking into account attackers, or I mean even potentially malicious insiders. And so a lot of the focus it seems right now is on continuous monitoring, real-time monitoring in the production environment. And over the next few years, I expect a lot of these practices that are still very nascent. That's where we're going to see the main changes in the production environment as supposed to the CICD environment, which has been the focus so far.

[00:49:23] JM: Okay. Well, it's been really great talking to you, and I look forward to seeing what developers in the future. Thanks, Mark.

[00:49:30] MT: Thanks for having me, Jeff. Pleasure being here.

[END OF INTERVIEW]

[00:49:41] JM: Join GitLab on August 26th, 2020 for GitLab Virtual Commit. It's an immersive 24-hour day of practical DevOps strategies shared by developers, operations professionals, engineers, managers and leaders. The attendees will hear from the US Air Force and Army. The GNOME foundation, State Farm, Northwestern Mutual, Google, more companies, many more companies about problems solved, cultures changed, release times that have been reduced. You can come and be part of the community that is just as passionate as you are about DevOps. You can register today at softwareengineeringdaily.com/gitlabcommit. Register for

GitLab Virtual Commit by going to softwareengineeringdaily.com/gitlabcommit. Thanks for listening and thank you to GitLab.

[END]