May 5, 2020 by Laura Shin

DeFi Security: With So Many Hacks, Will It Ever Be Safe?

Dan Guido, cofounder and CEO of Trail of Bits, and Taylor Monahan, founder and CEO of MyCrypto, discuss all the recent hacks in DeFi, how it can be made more safely and who is responsible.

We tackle:

- the Hegic security incident: whose responsibility it was to make sure the contract was secure — the auditor (Trail of Bits) or the team (Hegic) — what Trail of Bits was saying in its audit summary, and how to read between the lines of an audit summary
- how long an audit should be
- upgradeability: particularly around when more advanced technology and contracts interface with older technology/contracts
- centralization vs. decentralization: whether contracts can be made safely while maintaining adhering to the principle of decentralization, why Taylor would prioritize centralization and security, and how teams can create different levels of risk for users
- bug bounties: why asking what amount they should be is the wrong question
- the security threats posed by oracles
- and what a checklist for DeFi teams might look like

Crypto.com: https://crypto.com

Kraken: https://www.kraken.com

Stellar: https://www.stellar.org

Episode links:

Dan Guido: https://twitter.com/dguido

Trail of Bits: https://www.trailofbits.com

Taylor Monahan: https://twitter.com/tayvano_

MyCrypto: https://mycrypto.com

Trail of Bits presentation on blockchain security: https://www.youtube.com/watch?v=8lFK4jyAoKg

Initial tweet by Hegic calling the security issue a typo: https://twitter.com/HegicOptions/status/1253937104666742787?s=20

Hegic tweet saying, “It’s not a security issue”: https://twitter.com/HegicOptions/status/1253954145113038849?s=20

Trail of Bits saying it will no longer work with Hegic: https://twitter.com/dguido/status/1254260725431894020?s=20

Taylor breaks down the audit summary: https://twitter.com/MyCrypto/status/1254058121342803968?s=20

Molly Wintermute’s Medium post on requesting a week audit vs. three-day review: https://medium.com/@molly.wintermute/post-mortem-hegic-unlock-function-bug-or-three-defi-development-mistakesthat-i-feel-sorry-about-5a23a7197bce

Unconfirmed episode with Haseeb Qureshi on the Lendf.me attack: https://unchainedpodcast.com/haseeb-qureshi-on-the-unbelievable-story-of-the-25-million-lendf-me-hack/

Unchained interview showing Matt Luongo’s approach to kill switches and upgradeability with tBTC:

Discussion of the bZx attacks on Unchained: https://unchainedpodcast.com/the-bzx-attacks-unethical-or-illegal-2-experts-weigh-in/

Issue with Curve contract: https://blog.curve.fi/vulnerability-disclosure/

Compound bug bounty program: https://compound.finance/docs/security#bug-bounty

Taylor on "upgradeability makes things more insecure": https://twitter.com/tayvano_/status/1222564979657723904?s=20

Synthetix oracle incident, allowing a bot to profit $1 billion: https://unchainedpodcast.com/how-synthetix-became-the-second-largest-defi-platform/

Taylor's tips on how to get more ROI on an audit: https://twitter.com/MyCrypto/status/1254061500244713474?s=20

Tips to follow before getting an audit: https://blog.openzeppelin.com/follow-this-quality-checklist-before-an-audit-8cc6a0e44845/

Resources for security in DeFi:

crytic/building-secure-contractsGuidelines and training material to write secure smart contracts – crytic/building-secure-contractsgithub.com

https://consensys.github.io/smart-contract-best-practices/

https://forum.openzeppelin.com

https://swcregistry.io

https://diligence.consensys.net/blog/2020/03/new-offering-1-day-security-reviews/

Transcript:

Laura Shin:

Hi, everyone. Welcome to Unchained. Your no-hype resource for all things crypto. I'm your host, Laura Shin. Twitter fights, media posts, scammers, phishers, and

Unchainedpodcast.com to get a quick and easy summary of the top news stories every week.

Kraken

Kraken is the best exchange in the world for buying and selling digital assets. It has the tightest security, deep liquidity and a great fee structure with no minimum or hidden fees. Whether you're looking for a simple fiat onramp, or futures trading, Kraken is the place for you.

Crypto.com

In response to the challenging times, Crypto.com is waving the 3.5 percent credit card fee for all crypto purchases for the next three months. Download the Crypto.com app today. Download the Crypto.com App today.

Stellar

The Stellar network connects your business to the global financial infrastructure whether you're looking to power a payment application or issue digital assets like stablecoins or digital dollars. Stellar is easy to learn and fast to implement.

Laura Shin:

Today's topic is security in DeFi. Here to discuss are Dan Guido, Cofounder and CEO of Trail of Bits and Taylor Monahan, Founder and CEO of MyCrypto. Welcome Dan and Taylor.

Dan Guido:

Hey, there. Happy to be here.

Taylor Monahan:

Yeah. Super excited to talk about this.

Laura Shin:

Before we dive into the meat of today's discussion, can you each describe what you do in crypto and how you came to be involved in DeFi and / or security. Why don't we start with Dan?

Dan Guido:

myself and two other expert hackers in order to improve the foundation that we all build on. So I used to do just plain old code reviews for folks for many years, but in Trail of Bits we try to actually engineer software and build new solutions that others can use to kind of lift all boats. So we do a lot of work with DARPA and the DOD to build really advanced tools and advanced fundamental research. Companies hire us to build high assurance software for them on their behalf and then we also do really detailed product security reviews and training that help engineering teams build more secure software.

In the Ethereum space, this was actually out of personal interest. Or out of blockchain this was out of personal interest. We had a couple folks on the team that were as excited about the technology as folks that were working in the field. We saw this tremendous greenfield opportunity to come in and build the kinds of tools and techniques that other fields really ought to have adopted at their first steps but did not and that's what we did.

So now, in 2020, we have this massive suit of tools that people can use to build secure code and a vast amount of public knowledge that we've been able to communicate to folks that helps them build secure code. That's what we continue to do until today.

Laura Shin:

And Taylor, what about you?

Taylor Monahan:

I have a very different background. I started out just because I was in crypto, then I accidentally, as I say, built this wallet that became immensely popular and my security knowledge and the obsession I have towards everything that could possibly go wrong has sort of evolved over time. As I've watched things go wrong again and again and again and so MyCrypto is a wallet, previously I built MyEther wallet. These products are really interesting attack vectors but also there's a lot of unexpected things that happened as I was growing these products.

So today, really I'm just quite obsessed with how things go wrong, how we can prevent things from going wrong, what steps we need to take to improve on both the user side, on

Laura Shin:

Yeah. So over the last several months, DeFi has seen a number of security issues. It's funny because when you look at the discussions around all these attacks, they're just so new that usually people are even arguing about what to call them. I think obviously we can safely say that most of them are attacks or bugs. They're generally just ways in which the behavior of the protocol diverse from the intention of the creators. **So let's start by talking about one of the most recent one of the security flaws, which was on Hegic. Dan, you were actually involved in this one so let's actually have Taylor describe what happened first from an outside observer perspective and then you can jump in afterward. So Taylor?**

Taylor Monahan:

Yeah. So Hegic, the way it came on my radar was I was scrolling through Twitter, as I often do, and stumbled upon this tweet by them. I didn't actually follow them so it was retweeted by someone that said there's been a typo. You need to take this action. Warning, warning, warning, and then a couple other tweets and then obviously a whole bunch of replies to that original tweet with people kind of questioning how this happened, why it happened, what exactly is going on, etcetera. As I dived deeper into everything about this tweet in discussion, I realized that there were some really overarching problems with just everything about it.

So the first being that they initially called it a typo, which it obviously...while it may be factually correct, it's painful to watch people try to downplay issues like this. That was pretty frustrating right off the bat. That put me in a bad mood.

Laura Shin:

Yeah. And just to clarify and maybe I'm wrong but essentially this "typo" is that like if people kept their money in there their money would be frozen. Not like stolen but frozen.

Taylor Monahan:

and then they tried to clarify but they ended up saying it's not a security issue. It just ends up that everyone's funds are locked and now I was very involved in the parity multisig situation, which was I guess two-and-a-half years ago now, in which all the funds were locked and it was a huge thing and so to just kind of juxtapose those two experiences and then have someone be like, oh, it's not a security issue. I just found it preposterous, and I would say it went downhill from there.

Laura Shin:

Yeah. **A lot of people were tweeting about how they should call it a bug not a typo and spelling out what could happen to people's funds if they didn't remove them. So Dan, can you now explain how TrailBits was involved in the Hegic incident?**

Dan Guido:

Sure. So in line with my introduction, we try to remain as open as we can and work with everyone that wants help in the Ethereum, in the blockchain community. There are folks that are at the beginning of their journey building a product and there are folks that are partway through or at the end right before they're about to deploy something and there's always some guidance that we can provide. So I try to keep my door open when people show up and they say I have written something and I need your help to secure it. That's what the folks behind Hegic did.

They showed up. They said, look, this is a small project. We don't have a lot of money. We'd like you to take a look at it. What can you do for us? Over the course of three days, we found a large number of issues in a product that was at a very immature state of development, and we described those issues to them and said here are the things that you need to do to improve the security of this app.

Now, they took that, and it can be difficult sometimes. There's all these competing interests in the blockchain community around how you describe to your users that you are doing the diligence required to build something that others can rely on. Sometimes, that gets boiled down to a tweet-sized bite of information that we work with Trail of Bits or we worked with X, Y, Z firm, whether it's us or somebody else. There are ways to do that that is nuanced

software and we don't make any choices about how they built the software. They choose what development methodology they'd like to use, what build tools they'd like to use, how adequately they'd like to test it, the architecture of it.

We just come in and we try to do our best to make sure that it gets better after we leave. So instead, in this case, we provide those recommendations of here's how you should talk about your security process, and they didn't do that. They said, look, our code is all safe because Trail of Bits used it and we're launching it today. Now, we put out a summary document that said, hey, here's what the project with Hegic looked like. We did three days of work for them, which is a very small amount of work, and we found a large number of issues only two weeks ago. So from my perspective, this is a page-and-a-half document that includes a page-and-a-half of fairly negative information about the maturity of the product but most other people didn't read it that way.

Most other people looked at, well, Trail of Bits found some bugs, Hegic fixed the bugs, therefore the code is safe, which is really not the right takeaway, and there's some things that I can do to improve that but there's a lot about the community where the way that they are investigating what financial opportunities to provide their money to is kind of not producing the results they want. So there's a larger discussion here of like what are the factors that I should use to trust a given project and how can I interpret the information that's been provided to me about the safety or the viability of a given DeFi project. What is your question?

Laura Shin:

Well, I wanted to ask about something that you mentioned at the very beginning when you said you will work with a project of any stage because so based on the screen shots of the emails that the anonymous developer Molly Wintermute sent to you, this person wrote the letter Z for the word the and then numeral two for the preposition to and just I mean obviously I'm a journalist so the way the grammar and spelling and all the punctuation and all those things are very important to me. Just looking at that, that looked like a red flag even before receiving the code. It's like literally just the query itself to me, and I'm not trying to be judgmental of

Dan Guido:

It was so weird.

Taylor Monahan:

It is really, really weird. It's really off putting.

Dan Guido:

So why work with someone like that at all?

Laura Shin:

Yeah. Because for me it would be like this business relationship may not turn out super well. That is what I would think if somebody sent me a message like that.

Dan Guido:

That's totally correct. My thinking behind that is there are a lot of strange folks in the blockchain community. We've worked with somebody named Barata before, who ended up being an extraordinarily talented software engineer that helped provide input to enhance the quality of a product that we've created called Critic. We've worked with a pseudo anonymous personality over at MakerDAO named Rain. He would show up on video calls as a black silhouette and never went by anything other than those words. So sometimes in this community because of the privacy and very crypto punk, cyberpunk kind of approach people want to remain pseudo anonymous. They don't want other people to know exactly who they are. They kind of approach their work in this way.

So while, yes, it's really weird, it's really strange. I didn't want to slam my door on working with this person because they were, in my view, purposefully trying to obfuscate their identity, which is a thing that we've seen from other people.

Laura Shin:

Wait, but by using that language?

Dan Guido:

process all their text through in order to create something that's more difficult to fingerprint. You've seen this a lot back in the old days when there were hacker crews and everybody went by all kinds of different handles. You try to obfuscate any of the publications that you made by processing it with some kind of script to eliminate the writing style that you have so that people couldn't figure out who you are. There's all these techniques that come from stylometry and trying to figure out who Shakespeare was and whether he's this guy or that guy based on their published body of work. You could apply that to software engineering. You can apply that to text files that people read on the internet, and it's something that I have seen before of people trying to just avoid discussion of precisely who they are.

Laura Shin:

Okay. I don't know if I totally think that obfuscating someone's writing style is the same thing as switching out the letter Z for the word the. Anyway, I don't want to get too far down that. Taylor, what did you want to say?

Taylor Monahan:

So yeah, so when I first saw her, I guess, writing style I was taken aback by it and had the same sort of feelings as you. I think that there's a couple different things going on. One is that, yeah, the crypto space is just weird and so when you see something like this it's not as weird as if you're in a normal corporate environment and you get an email like this. And the other is that the cypher punk mentality and the value that these sort of cypher punks can provide makes it so that sometimes you'll give people the benefit of the doubt when they don't necessarily deserve it and would never get it in another sort of industry or situation, and I think that's definitely what happened here.

Dan is not the only one who has brought up this obfuscation, this on-purpose obfuscation and one thing that lends itself to that theory is that Hegic has a lot of writing out there that is not written like this. The Hegic Twitter account does not write like this. The whitepaper is not written like this. The website's not written like this, and so I'm not exactly sure why this personality, Molly, every time she writes there's the Zs and the twos when she's

say on that.

Dan Guido:

And those are things that we've looked at, too. Obviously, we've been approached by people that are building pyramid schemes, and it's really easy to figure out when somebody is fraudulently trying to manipulate users in that way. But from the kinds of things that Hegic was talking about from the kinds of things that were documented in the whitepaper, there were friends of mine that were following their Twitter account already, so they kind of had at least some items here that meant, hey, maybe this is something that ends up becoming important in the DeFi space and maybe I should look at them despite this weird interaction that I'm having by email.

Laura Shin:

Okay. **Well, one other thing actually that I did also want to ask about, so Taylor did such a great tweet storm dissecting the audit summary where she basically says all this is written very professionally but here's what they're really saying. Correct me or Taylor if this was wrong but she was like, oh, Trail of Bits is saying they didn't even do basic arithmetic correctly. They didn't even do the basic thing of having documents. So I just wondered around things like documents, would it ever make sense for an auditing company to state that potential clients need to meet certain requirements before they can have an audit?**

Dan Guido:

So I think there's never a point where it's too early to engage with a security professional. That just kind of brings up this question again of are there people that I should tell to go away from my queue of folks that are asking for help, and I don't think that's the answer. However, there are a couple things here. So first off, the fact that we found...yes, we found all these critical issues in basic arithmetic. We found 10 issues that essentially could have stolen everybody's money two weeks before this project was going to deploy. They fixed only those specific issues and they didn't address any of the root causes of any of them.

That there was no substantial foundational improvements made to that code beyond patching individual lines of code. So that's something that you need to understand about these security reviews is that they're usually focused on, hey, we spent X amount of time looking at the code and we found Y issues. If you found a lot of issues in a small period of time. It doesn't mean security improved dramatically. It means that the code is probably filled with bugs. Like big number is bad, and I think that most of the community thinks that big number is good.

Taylor Monahan:

Yeah. I think there's a couple of really important things that this has brought to light and one of them is the way that I read audits and I look at teams is very different than a lot of other people look at them I guess. The way that I look at them is not what does this audit say about the code but what does this audit say about the team or how they're approaching the code or how they're approaching DeFi.

So when I was reading the review I was like this indicates to me that they're not really taking much seriously. They went into this audit pretty unprepared. There were some pretty basic things that they could have fixed before sending it to audit, etcetera. People on the technical side kind of tend to miss how human all of this is, and people that don't have any technical experience don't realize that they can read these audits and apply sort of like the softer skills or the culture takeaways even if they don't understand the literal technical underlying stuff and that's one of the, I think, the biggest missing pieces.

It's like you have to have both sides. You have to fix the bugs but you also have to try to understand why they even got in the situation in the first place. Why is this code being given, basically handed to Dan in Trail of Bits with a chunk of money being like we're ready to go live except we want a third set of eyes on this? And if you send it over in that state, that alone to me is a red flag. In theory, you thought it was as good as it could be.

Dan Guido:

To push back on that a little bit, I don't know what they're planning to do when they're sending me the code. I would have thought that a reasonable thing to do with this code

methodology where there are many more steps they have to take until they release it to main net. But instead, what they did is they shrunk all that down and said it's ready to go. Ship now. Now is when we're going to do this. I think this would have been great if they just said we've got some feedback about the maturity of our development. We've gotten consultation from experts. They said that the code is not great. They listed out all these things that we should do. Let's show it to users, get it some testing and continue to work on it.

That would have been the perfect use of an engagement with us but that's not what they did.

Laura Shin:

Wait. **I guess maybe I just have a misconception around what role an audit should play in any kind of security. If I sort of compare it to the way a magazine article gets published or something, I would imagine that the audit would be one of the last steps where you would get the most bang for your buck if you put forth the best effort you can, get it as ready as you can, as close to perfect, as close to launch as you can and then at that point have somebody come in from the outside.**

Dan Guido:

Yeah. Please, don't do that. Do not do that.

Taylor Monahan:

I'll just point out before I let Dan finish because he's going to be right but I just want to preface this by saying that the ideal way to engage with security experts is not how anyone's doing it right now. That said, Dan's going to tell you how it should be done.

Dan Guido:

Yeah. Sure. So a couple things here, before we get too far away from the point, I do want to say what Taylor brought up about the context around the code and kind of the organizational behaviors and their own maturity of dealing with security stuff is a really important thing for users to understand that I don't think shows up in many of these PDFs that come from vendors like mine. We try to do a good job at that. We always list out long-term

that, and we do so we're good in that respect, but I think we could do much better.

So some of the things that we've discussed internally since this Hegic thing kind of blew up is ways that we can provide literally a color-coded graph around the maturity level of various controls of projects in the DeFi space and the blockchain space. This kind of takes a lot of inspiration from the way that we do threat models for companies, which is a different kind of service, to get back to your original question now. How should people engage with a security company. They should understand a little bit more from a strategic level where their risks are. One of the ways that people do that is they use things like threat models.

So threat models are a technique to understand what data you currently collect and manage and process, the sensitivity of it, the components that do the processing and the requirements of those components to properly protect it. If you have an understanding of that stuff early in your development cycle then you've got a set of guardrails that make it much harder for you to get into situations where you inherit way too much risk, more risk that you're capable of mitigating.

Examples are like if you engage with a security professional early, there might be ways that we can discuss with you your goals and then within the context of those goals help you avoid manipulating low-level solidity calls in order to achieve them. Because manipulating low-level solidity exposes you to a vast amount of risk that maybe you can avoid. Then there are hundreds of bugs that will just never enter into the code base at all. If you wait until the end and you've gone out on this limb and you've built all this code and that's the first time that you're exposed to a security engineer, there might be a lot of cases where the code's been engineered in a way that is fundamentally unsafe and requires re-architecture. So there's no way I can secure a code base at that point. All I can do is point out all the things that are bad.

Laura Shin:

So basically there should be multiple engagements. Is that what you're saying?

Taylor Monahan:

perfect example of this is like when we were first putting together the first version of MyEther Wallet and sending it out into the world, we had assumptions. We wanted to make this tool whatever. We weren't really thinking that it was going to blow up but even down the line we never engaged with someone who did this for a living, who really, really understands security.

Fast forward to 2017, all of these things came out of the blue and hit us upside the face. The phishing attacks, the malicious browser extensions, on and on and on, all of these attacks. If we had talked to a security professional at any point before that point they would have said flat out, no exception, don't put private keys in the browser. The browser is unsafe. There's all these different ways that people will attack you that you can't control. BGP, the underpinnings of the internet, all of this is insecure. Instead, because we didn't, for various reasons, we basically built an insecure product that by the point we realized how insecure it was, it's really hard to take steps back and move away from it.

I think that's a bit of a more accessible example but this same exact thing applies to DeFi products, to smart contracts, to pretty much everything. You don't want to get too deep into it before you realize that you're going to have to change the entire nature of your product or your system to ever be secure.

Laura Shin:

One other thing I wanted to ask about because this was also a point of dispute, basically it looked a little bit like Molly Wintermute wanted maybe like a week-long review and you guys were saying a three-day review should be sufficient. So what amount of time would you recommend teams seek for an audit or I guess there's multiple audits at different points in their project.

Dan Guido:

Yeah. So I think part of the issue here is that we keep using the word audit as if it's this fundamental scientific process where we can eliminate all the bugs from the code. Taylor and I are both of the opinion that instead this is like a divining rod that lets us figure out where hotspots are and whether there's an underlying issue that needs to get

good coverage on the code, and what we found was that the code was bad. That was a sufficient amount of time for us to understand the current state of the project.

So in those three days, we found 10 critical bugs that allowed us to steal everyone's money and manipulate all the things that you thought you could depend on. That is not a great result and we only needed three days to get there. So the extra two days for a full week wouldn't have told us anything different. It would have been a waste of money, in fact. So they already had now a list of things to do, really what a security vendor like ourselves is trying to provide to people is a backlog of activities and investments in security that you need to make. So we filled up that backlog. There are now a half dozen or a dozen different things for you to do, and we want to try to provide the most information, guidance to you in the least amount of time, which is why I'm not going to oversell somebody on a project when I know that I can provide the results they need within a smaller time period. Does that make sense?

Laura Shin:

So it sounds like you're saying, yeah, that they were looking to you to fix all their problems or to kind of just...whereas you're saying that really the responsibility lies with them and that, yeah, you can point out the ways in which maybe their process or their culture around the way they're billing this is going to lead them into trouble but you cannot be the ones responsible for the security of their project.

Dan Guido:

Yeah. I mean they've been working on this for weeks, months, years. There are many people on the team and just by the virtue that I looked at the code for three days doesn't mean that I'm ultimately responsible for the security of their entire company and product. That's just the bottom line. There's a lot of other questions that you could ask, too. There's things that are outside the actual code that determine the security of the product that I'm sure Taylor knows about just as well. Things like the owner privileges in the DeFi space. People are obsessed with that. How decentralized is the application?

Things like the Oracles that are providing feeds of information that the DeFi application might make decisions

the code that get made after a review is done. There could be things about monitoring. Do you even know when things have gone wrong in the future? Those could be things that firms ask me to help them with. I can help you build a process around security monitoring so that instead of the public finding out that you've been hacked, you find out that you've been hacked first and can take some kind of remediation, maybe immediately issue a contract migration that saves some portion of your user's data or money.

There are many things that can go wrong, and it's really on the owner of the DeFi projects to fully understand what those things are, and they can use our help when they ask for it, and I'll provide them the best guidance I can. All the best practices, all the new solutions. We'll bring in all the expertise we can to accelerate them but ultimately it is their responsibility to build a secure product.

Laura Shin:

Yeah. This is actually a perfect moment to take a break because you basically listed a whole bunch of things that I'm going to ask you about in the second half of the episode. So here we'll get a quick word from the sponsors who make the show possible.

Syndicate lite, more frequent events with slightly lower discounts for the hottest coins. The first event is offering bitcoin at 25% off on 4 Feb! Sign up on the Crypto.com Exchange now to participate.

Stellar

The Stellar network connects people to global currencies and assets. Stellar lets you make near-instant payments in any currency with anyone, anywhere. It's an open blockchain network that acts as payment rails for applications and institutions around the world and designed so that existing financial systems can work together on a single platform.

Transactions powered by Stellar are low-cost, transparent, and fast, saving both businesses and end-users the time and money associated with traditional payment networks. With Stellar, your business can issue digital dollars or exchange existing fiat currencies without the need for complicated smart contracts or new programming languages. Its robust documentation, toolkits, and multi-language support let you quickly integrate Stellar into your existing products and services. Learn more about Stellar and start building today at unchained.stellar.org

Laura Shin:

Back to my conversation with Dan Guido and Taylor Monahan. So let's actually just now turn to another recent pair of attacks. These involving IMBTC on Uniswap and then also on the dForce protocol's Lendf.Me platform. Hopefully, the audience here caught my Unconfirmed episode with Hasseb Qureshi on these incidents because it was actually really, really fun chatting with him and definitely it's a crazy story so you should check that out. Essentially, both of these attacks were caused by this ERC 77 token, which is sort of like a more kind of upgraded or advanced version that has basically just other kinds of functionality that ERC 20 tokens don't have.

However, if an ERC 77 token is used in an older smart contract that does not recognize that then an attacker can perpetrate a reentrancy attack using that token. So I was wondering how you guys thought about situations like this. How do you think DeFi should handle situations

Taylor Monahan:

This is actually the thing that scares me the most about smart contracts in general. I have no doubt that at some point we will get to the point where we can write secure, solidity, or whatever language it's going to be. I have no doubt that we can get the community onboard with understanding what makes a secure team, etcetera, etcetera. But when you think about the fact that there are all of these systems, there's the dForce system or the ERC 777 system or the Unisoft system or whatever it is, you can make all of the pieces secure, and you can have them implemented by good teams that are security minded and then you combine two of them and everything goes out the window and now there's problems.

When you think about just how many different combinations there are and the fact that you can combine two or three or 10 of these systems, it's really hard to imagine on a purely technical level, there's no way to ever have the system as a whole every single possibility, every single combination, there's no way that it's ever going to be perfectly secure.

Dan Guido:

I have a little bit of a different take on this one actually. I wonder what you think.

Taylor Monahan:

Okay. Go for it.

Dan Guido:

So in the Uniswap dForce case, they were affected by the world's most well-known bug class in Ethereum. They were affected by reentrancy.

Laura Shin:

Made famous by the DOA, in case people don't know what that was.

Dan Guido:

It's incredible. It's so funny because up until this point, since the DOA, there hasn't been a really exploited reentrancy

harm are somewhere else. And now all of a sudden in 2020 we have a reentrancy that's used to steal real money. That was the most surprising part of this to me, and when I think about it a little bit, why weren't they aware of a basic reentrancy flaw in a set of contracts that's got a lot of eyes on it but has actual development teams that are trying to do their diligence to build it.

And you look at the technologies that are being used. In the Uniswap case it's Viper and the tools for a lot of secure development and bug finding and vulnerability discovery are written specific to solidity.

Laura Shin:

Wait. Dan, just go back. So Viper is... what is that?

Dan Guido:

So there's choices that you can make around what programming languages to write smart contracts in. Most people choose to use Solidity. Solidity is filled with footguns. There are many ways that you can step on sharp objects and end up really hurting yourself with Solidity. So there's a community of people that have developed a new language that looks a lot like Python called Viper. Now, Viper, while it has a lot less footguns, a lot less sharp objects all over the ground that you could potentially step on there are still some fundamental things that you need to do correctly and avoiding reentrancy is one of them.

So the problem here is that a lot of the best tools in the space for detecting basic security flaws like this have trouble working with Viper. So the issues with adoption here of those tools may have created a scenario where it was more difficult to find in a Uniswap and IMBTC kind of scenario just because they've chosen to use different sorts of tech. Now, on the other hand, the dForce folks are in a different position because they did use Solidity and that's simply a question of there is a check box, yes / no answer, that you can get of have you evaluated your code for known flaws and ensured the absence of them and for dForce the answer was, no, we have not because this would have been detected immediately by any kind of off the shelf security scanner that exists in a space.

Laura Shin:

this last summer in July. So it's been known for quite a while. But actually, one other thing I wanted to bring up about this is that one thing that Haseeb said was that for instance so dForce had copied Compound's code but what he was saying is the reason that this issue didn't come up on Compound is because Compound knew about the issue and made sure that no ERC 777 tokens were put on Compound but that happens because they have kind of more centralized control. **So I feel like there's this tension between the decentralization philosophy and then having good security. How do you guys think about that?**

Dan Guido:

Yeah. Common thing in the DeFi space, there is a lot of risk around composability, which is I think the word we've settled on to describe all these emergent behaviors and potential interactions between things that happen on chain and the security risks that come from them. When we work with projects, we've worked with Compound as well and a lot of the way that you have to approach this is by white listing the behaviors that you have studied well enough that you trust and slowly opening up the ability of your contracts to interoperate with other stuff. So if you don't fully understand all the repercussions of working with arbitrary ERC 777 contracts then maybe you should wait until you're fully clear on what that means before you allow your contracts to do so.

Now, that's like one strategy but at some point composability is unavoidable. I don't know. An example, you could buy insurance that's been collateralized with DAI and then there's three systems that all interact with each other. So at some point, there's no real way that you can avoid that compatibility so it's really everybody's responsibility for ensuring that the contracts and systems they use are...that the interactions they have with them are safe. It's something that I haven't seen my DeFi projects fully internalize where I think most projects in the space still depend on outside experts like TrailBits or someone to come in and advise them about what's going on and what they should pay attention to next and new objectives they should build towards.

project where they had an arbitrage contract on chain that was abusing their app. Investigating that issue required them to identify the arbitrage contract, download the binary code, reverse engineer it with one of our tools and then deeply understand the way that it as abusing their work. That's something that I think DeFi projects are going to need to come to terms with. They really need their own deep understanding of these issues to deal with them in the future.

Laura Shin:

Right. Yeah. That still is, I think, a more centralized model but also this kind of is also related to the upgradability thing, so the way I asked the first upgradability question was just about when there's advancements in technology then what do you do especially if your project at that point is more decentralized and you have less control **but then another question is just like how should each system be upgraded?** I had this discussion with Matt Loungo about tBTC the other day where he...at different points in the interview, one time he was like, oh, we're going to set it and forget it kind of attitude and then later he talked about the next version and he was like, oh, yeah, well actually there probably will be a V2. But yeah, I just wonder how do you...I can't imagine...so let's say DeFi becomes a thing. **Ten years from now, we're not going to be using the current smart contracts, right**. **But yet how do we get from here to there while keeping in mind all these different principles like decentralization and security and upgradability, etcetera.**

Taylor Monahan:

Yeah. So the way I look at it is right now the biggest threat is we are writing bad code. We are creating insecure systems and so in the short-term I would prioritize centralization and security over decentralization. That's not to say that we should just forget about decentralization and not have it be sort of part of our goals or our philosophy but right now the worst things that can happen can either be mitigated or eliminated by having just a kill switch. I really lived through the DOA and I can say that everyone who was there is in the same mindset because we've watched what happened when you try to fully decentralize everything and you're not ready to.

Oh, wow, but I mean I'm sure you're aware you just made a controversial statement.

Taylor Monahan:

I mean I do get it and that's the thing. It's definitely a conflict within me because I'm building on Ethereum. I love decentralization. I love what it empowers, but in the short-term we're never going to be able to get there if every single contract is the DOA and it just blows up and everyone loses their money. So there's some really interesting ways where you can strike a balance in the short-term and then as the system becomes more secure and more mature and you have confidence in it you can ramp down. Matt Loungo obviously, he has one approach that's a bit too decentralized upfront for my taste. I've talked to him about this. But just as an example, you could have a smart contract where you have a big red button where if something goes wrong you push the button and it stops everything except it allows for one function that allows the user to withdraw their money.

So now with a hacker or a flash loan or an arbitrager comes in and starts screwing with your system in a negative way you can prevent them from doing that. You can prevent the bad things from happening but you don't necessarily lock the user out. They can still go and withdraw their money and you can also do that in a way where the user can withdraw their money but you can't, etcetera, etcetera, etcetera. So these are the things that I think in the short-term we should definitely actually be encouraging because if everything blows up and ever single project basically launches huge fanfare and then everyone loses their money we're never going to get to a point where any of this stuff is actually useful. So baby steps, please.

Laura Shin:

Yeah. **There's been a lot of hacks but one thing I just wanted to ask was when you said the user should be allowed to withdraw their money but you can't. When you said you did you mean the developers of that protocol?**

Taylor Monahan:

Yeah. Exactly. So whoever...I know this is almost a meme at this point but the decentralization is a spectrum, it really is

developer making a centralized decision to turn it off but that doesn't necessarily mean that you have to be able to turn it off and steal everyone's money as the developer. You can have a system where a centralized party can turn if off but they can't touch the money, they can't withdraw the money and still allow, in a decentralized way, each individual user to withdraw their money from the system.

Laura Shin:

I find that idea really interesting because basically what you're doing with that is you're making the risk for different levels of people in the system different. So if you're building it then their risk has to be higher. They have to put more effort into making it secure but for users their threshold is a little bit lower and by the same token they have more ability to go in and out.

Taylor Monahan:

Exactly. And we have not seen... I don't think we've seen a DeFi-specific product that has exit scammed or taken advantage of the decentralized mechanisms to steal everyone's money, whether that's either a team pretending to be good and they're actually bad or a hacker like abusing...I lied. There have been hackers that have abused the admin functionality of a smart contract but when Dan mentioned earlier threat modeling, is the team itself good or bad? Are there attackers on the outside, coming in from the outside attacking? Are there users that are inadvertently doing bad things on accident or on purpose. There's all these different parties.

You do have to be aware of them. You do have to try to protect against them. It's never going to be, especially in the short-term, perfect and secure against every single party and that's why, for me personally, again prioritizing the safety and the pause buttons and those types of tools do that.

Laura Shin:

I just wanted to ask you guys about one other thing that isn't exactly in your wheelhouse but I was so curious to know your opinion. So with the dForce attacks, they did call the Singaporean police on the attacker, and I just wondered in general, do you think the traditional legal

protocol who would be responsible or how would that all work?

Dan Guido:

You're right. That is a little bit outside I think our area of expertise, but if it's an option for you then I don't see why you shouldn't take it. There are two things I want to address about what Taylor mentioned. The upgradability conversation doesn't just affect the security of your product. You can also think about your product may be safe today but another contract on chain could upgrade or change their behavior and now their interactions with you are unsafe and that's where the whole flash loan thing comes in. There have been contracts that have been deployed for weeks or months or years and this changes the entire kind of threat landscape, all the bad things that can go wrong are suddenly much more severe and much more likely to occur and it was through no code change of yours. Your code did not change a single line but things outside of you did. So that's other things that you need to be aware of and have an ability to respond.

And I think empirically right now the level of decentralization in the DeFi space is very low. You can go download all the code for all the DeFi apps and run it through Slither, our static analyzer, and you'll see all the owner privileges that just drop out and it's extensive. I don't think anybody right now, very few people are really achieving that ideal goal of being fully decentralized, and I think that's okay. I'm with Taylor on this 100 percent. You have to take baby steps to getting there and it's going to be a long road.

Laura Shin:

Yeah. Well, since you mentioned the BZX attacks, let's definitely talk about those. I guess actually something that interested me is what you just said. You kind of sort of call out the flash loans as one of the issues but actually somebody else that I interviewed, Lev Livnev, when I had him on the show he was saying that for the BZX attacks he felt like they weren't necessarily the culprit. Obviously, they made it cheaper to make an attack, but he felt that really this was more like actual bugs in their code. So I was curious to know, do you think flash loans are a problem because

Dan Guido:

Yeah. So there's some nuance there. There was a specific coding flaw in BZX that allowed this attack to happen. They had a short position that should have been closed because it was undercollateralized, but it wasn't. That's the bug. However, the ability for somebody to exploit this became significantly easier because flash loans were a thing. So what I think most DeFi projects need to understand is the bar has now been raised. Issues that were low severity before are high severity now and that it's insufficient to only focus on a couple of thing that a firm like Trail of Bits reports to you that you actually have to go through and fundamentally address every issue.

This gets back to how do you actually secure a DeFi project or what is the process for securing a smart contract at all and ensuring that you're not exposed to known attacks is great but at some point you have to have a deep understanding of what your own code is supposed to do and be able to prove that it operates the way that you expect and that's defining security properties and testing security properties during development. That's like the next layer of a pyramid that I visualize of application security maturity where a third level might be all the token economics and the incentives that you've created, which is just a whole other thing that very few people have a handle on.

Taylor Monahan:

Yeah. I was about to say in addition to all the technical issues that we should be scared of, there's the whole thing where financials and incentives and economics and tokens, when you start thinking about that, those are attack vectors. If your token economics don't ensure that everyone is making money in the way that they expect to, bad things could happen. It may not be as drastic as the DOA but if you're promising a sustainable business and you're actually losing money every single month, that's an unexpected behavior and I think we are going to see way more of that come to the fore front as these more and more DeFi projects start launching.

Laura Shin:

the BZX attacks where the attacker was unhappy with the amount of the bounty offered, which was five thousand dollars, whereas with Compound bug bounties range from as little as 500 all the way up to 150 thousand. So I was wondering, how should protocol teams determine what amount their bounties should be. What do you guys consider fair? How is that determined?

Dan Guido:

So I don't think the conversation is about the bug bounty dollar amount. There are some people for which the dollar amount is not a thing. They don't care. There are good people and there are bad people in the world basically. There are some people that are going to do things to screw with you and there's nothing you can do to convince them otherwise, and there are other people that are good people that just want to help you and they're very receptive to any sort of assistance or acknowledgements or thanks or money that you provide to assist them. What you want to do is you want to make sure that all those good people that are out there that are willing to communicate with you are kind of incentivized and it is easy for them to contact you and get those issues fixed.

You don't want them to not know where to go, to end up tweeting about it, to end up putting it on Reddit or wherever else. You want to make sure that you actually hear all the things that people have to say. So providing that free flow of information is the most critical thing for a business bounty program, and that means describing things like safe harbors where you have language on your page somewhere that says here is how you can skip the support queue. You don't have to email support at whatever and create a Zen Desk ticket. No, you can reach our security team directly. If you do so, we won't sue you and here are all the different ways that we won't go back and harm you. It is safe to tell us things. So that's really important.

Taylor Monahan:

I'm with Dan on this one as well. The bug bounty number, there's all sorts of philosophies on it but that's the least important bit. The most important bits are everything else because if you think about typically they're called gray hats.

them on your side. You want them to be white hats for you. So the ways that you can do this are essentially by not pissing them off and by making it very easy for them to get you information. Both of those are insanely important because you can imagine that if someone either accidentally stumbles upon something or is hunting for something and then they try to get ahold of you or they try to share it or they try to figure out what this piece, how it connects to that piece or whatever it may be, every single one of these steps is going to irritate them more and more and more.

It doesn't take that much to piss people off on the internet, and again, if the person is somewhere in between perfectly good and perfectly bad they may either just not disclose it, just give up and be like screw this or they may be like, hey, I don't know what the heck's going on but here's this huge exploit and just dumb it on Twitter. We've seen this again and again and again. So bug bounties, you should have the page, you should encourage people, you should give them all the ways to communicate with you. You should respond to those really, really quickly and professionally. You should have sort of your security information everywhere. Dan has a repo called The Blockchain Security Contact List. If you're not on that list when, say, Sam Sun finds an exploit he has to go into Telegram and be like, yo, anyone know how to get ahold of "X" team and then we're all sitting there going, Jesus. Again, Sam is this example of pretty damn close to perfectly good but most people aren't going to be sitting in a Telegram with a whole bunch of blockchain people and ask for a contact and then get an answer in two seconds.

Dan Guido:

So the other thing to say here, too, is it shouldn't...so if this person was truly motivated by the amount of money that was being offered, that person still should not be able to ruin your day by virtue of them tweeting about some bug in your contract. You can't depend on the fact that the bug bounty exists, that no zero days will ever get dropped on your system. So this goes back to that security response discussion we had a few minutes earlier where you need to have processes and procedures in place where you know what to do and you can safeguard people's money and you can take appropriate steps to respond to issues when they come out. Just because you've got a bug bounty doesn't

Laura Shin:

What does that process look like because with BZX there was yet another issue here later on it was revealed that one exchange had actually previously notified them of a different vulnerability and then took issue with the fact that BZX did not pause their protocol during the 16 hours in which they created and deployed a fix and so user funds were basically vulnerable during that time and there was a similar incident with Curve and during that time, they kind of couldn't figure out should they alert people to what's going on because if that happens then black hat hackers could take their money. Their contract actually didn't have any kill switch or upgrade ability. So they ended up deploying a new version and the new version had the fix but they didn't disclose any of that and then they kind of waited until most people migrated over to the new contract and then afterward they announced it. **So just curious, how do you think teams should handle bugs when they find out about them?**

Taylor Monahan:

It all depends on the situation. If the very first Parity...that was the huge conversation because the people that were discovering this situation were discovering it all based on public information. We were all just looking at the chain, which means that anyone else could discover it. Of course, we're not the Parity team and we're also not able to put a kill switch or anything. Again, this is one reason I'm a fan of kill switches because if you can kill it, it takes a lot of the options off the plate. Yeah. Striking that balance between not telling people and keeping thing secret and the flipside of telling everyone and knowing that everyone also includes people that are just going to exploit it and steal all the money, it's a really, really tough position to be in. This is why kill switches should exist because the takes that decision away.

If Curve could have said, oh, shoot, and then just pressed pause they wouldn't even have to go down that path because once you're going down that path there's no right decision, there's no good decision. You're in that situation where what's the least shitty position?

Dan Guido:

company to deal with those unforeseen circumstances. So what are the set of things that could go wrong and how will we react to them when they do? You're not supposed to figure that out on the fly. You should ideally have that in place while you're developing the product and there are many choices that you can make some are going to work out like Curve in their case maybe withholding a little bit of information but then clearly explaining it after they took actions to secure their users' funds. It might be the right decision for them but it could be the wrong decision for somebody else. I don't have any specific concerns about what they did. I think kind of the ends justify thing means a little bit in that case since you safeguarded people's money, but it really, really depends on context.

Taylor Monahan:

Yeah. Exactly. And the thing is, is that with all the situations where people withheld information and then revealed all shortly thereafter, I don't take issue with that. It's when they don't reveal all or when the information is so available yet this core group is denying, denying, denying. Once the swing has swung you have to go all in and make sure that people do have all of the information.

Laura Shin:

Now, let's discuss oracles. That's an area that's pretty susceptible to attack and they can also be ripe for manipulation. Last summer, there was an oracle for the price of the Korean Wan on synthetics that was just incorrect and somebody was able to obtain a billion dollars in profit with their bot. Yeah. Exploiting that. So I just wondered what your opinion was on oracles. Is it too early to have reliable ones and if not, are there any particular characteristics that give you more confidence in certain oracles versus others?

Dan Guido:

Yeah. This is just a huge discussion around the security of your code doesn't just depend on your code itself. You also have to consider the environment around it, the environment that it operates inside. So when I'm looking at judging the reliability of a DeFi project, some things I really want to know are how many oracles do they rely on and how many would have to be untrustworthy for there to be

my pyramid little thing where you've got your known vulnerabilities at the bottom. You've got your application specific stuff in the middle and you've got your economic model up at the top. This is definitely a blend of steps two and three here where you need to actually model that behavior and think through what could possibly happen.

There are some tools that you can use to model that that are already available but they're not purpose built for this task. You can use tools that come from TrailBits like Echidna and Manticore, which are essentially a little EVM runtime written in Python and written in Haskell that you can use to evaluate your contract with different environmental data being provided to it, but they're really more meant for finding more code security related issues and less about providing this feedback on the behavior of your code in response to all these weird oracle things. So I think that's a part where the tooling and the knowledge could get a lot more mature over the next few months or year and it's certainly an area where it's needed as this incident shows.

Taylor Monahan:

Yeah. And I'll just point out that a lot of the exploits that have been responsibly disclosed in the last two, three months have also surrounded either oracles directly, manipulation of the price that the oracle is getting the information from, like that even played into the BZX incidents as well. And yeah, again, there are so many different ways that these systems can be outright attacked or have an accidental bug or be manipulated. In any of these unexpected behaviors, you have to think about them upfront because otherwise they're going to hit you hard and you're not going to know how to respond. You're not going to be prepared, and that's why I think the overarching theme of this conversation is we're not mature. We're not ready for this. What do people need to do? What do the teams need to do? What does the community need to do to get a little bit better?

There's all these little things that they can do to prevent bugs and there's these tools that you can use to write better Solidity or check your Viper or whatever it is, but at the end of the day, there's so much going on that at least for me what I look for is a team that is really obsessed with security, that's paranoid, that understands that bad things can

super paranoid, that's the best hope because there's so many unknowns.

Dan Guido:

That's my hot top for figuring out if a company's got a secure product, too, for non-blockchain software. I always just pop them open in LinkedIn and I search for security and their company name and I see how many people they have working for them that actually have a responsibility to secure their company. If it's zero then I know that this whole thing is probably a garbage fire but am I okay with that? So it really goes back to the same thing of does anybody working for this DeFi project have experience that would indicate they know about security stuff? Did they work in traditional finance at some point to have that sort of background? Or do they have a past history of development or publications or at least public communication that they understand what they're in for as they're building this product. If no, that's a serious concern and that's really the underlying most fundamental concern that I could have about the project. Do I trust the owner? That's a question you can ask from multiple angles. Do I trust them not to run away with all my money and do I trust them to actually do what's responsible to protect it?

Taylor Monahan:

Yeah. Exactly. And the answer to that is not ever, well, Trail of Bits audited them, therefore I trust them. And that's what I think the fundamental, all of these disagreements about audits and what they are and what they're not, why the whole thing is missing the bigger picture, which is there's no one thing that any team can ever do to be perfectly secure and so throwing it on Dan's head when something goes wrong is preposterous because you're not asking the right questions in the first place. You are not asking the right questions.

Laura Shin:

So I might not be asking the right question here because I actually asked...so normally, for certain episodes I don't tell the guests what the questions are but here I did ask Dan and Taylor to come up with maybe a checklist of things that they think DeFi protocol teams should do before launching a protocol because I wanted them to

thought my question didn't make sense. So curious to know what your answers are.

Taylor Monahan:

Dan, you want to start?

Dan Guido:

Okay, yeah. Sure. So I thought about this a lot over the last few days because of this incident with Hegic where it can be difficult for an outsider to understand the level of maturity of a project and that's really what we're trying to get at is what are the long-term steps that someone should take to end up arriving at a secure product and how do we evaluate those, how do we communicate those, and what are the important steps within them to have actually taken.

So on one hand, we have a set of critical controls that are necessary for DeFi projects to have. They have to have access controls. They have to deal with numbers correctly. It's kind of important. The degree of centralization or decentralization. Their documentation and specs. The kind of key management that they use. Their security monitoring, the level of testing that they've gone through. Those are all kind of indicators and what I think we're planning to do for our reports in the future is rank all those critical controls from weak to excellent where each of them, there's no overall rating. There's no like, hey, this is safe. At the end of the ay if you get five out of seven then you're good but it'll at least provide some information from our team in our expert view where we think they are in terms of building a defensible system.

Now, that's one way to take it and that's sourced from the threat models. There's another ancient web kind of document that I love to cite. So if you go back to the year 2000, there's a guy on the internet named Joel Spolsky, kind of a famous guy. He created the FogBugz system, one of the best bug tracking managers that people had before like GitHub and GitLab were about. Created Trello and has kind of just been a software engineering leader for many years. He came up with this thing called the Joel Test and it was a set of 12 yes or no questions that you could ask in 30 seconds or less to figure out the maturity of a development team building software.

model, CMM are a kind of really rigorous way to evaluate if a team builds good software, and he managed to simplify it down to a 30 second yes / no exercise. So what we've done is we tried to build that same thing for Ethereum, and we could call it the Dan Test but it's also kind of the Dan / Jocelyn Test since he came up with a lot of it with me. But there are some basics here like can you compile without warnings on the latest compiler that you're using. Do you important third party libraries from a package manager and track their versions? Have you located and documented every privileged operation in the system? If you can't say yes to those questions then you're probably not ready to go. So I have a big list of those. I'm going to publish them all next week.

Laura Shin:

Oh, great. When you do that send me the link so I can put them in the show notes.

Taylor Monahan:

I'm so glad you're doing this because it's really tough and this is what I think that I asked on Twitter two weeks ago now. What are the things that every developer should do before having 25 million dollars in their contract on main net? What are the big red flags? There's a lot of really deep, in the weeds type things that I think are really, really important. But it was actually interesting because some of the responses were very different but also really enlightening. So one thing that came out of that conversation was if someone doesn't have an audit that's a really big red flag. If they don't get anyone to look at their code, that's a red flag. Just because they have it on it, it does not mean that they're secure, it does not mean that they're ready for main net. It just means that there's not a red flag in that area. It doesn't put a green one there. It's just not a red flag.

And then some of the other ones that I think were really interesting were around the teams and the people and how sort of...like how much effort and time they dedicated to the things that weren't the literal code. So a lot of teams obviously love to focus on the code, they love to focus on the product. They want to build this awesome system but did they spec out the project before starting to write that code

white paper like a marketing piece or is it actually a technical document that dives into all the different situations?

Another really interesting one that I can't necessarily call it a red flag today because not a lot of people do it but certainly would allow me to have more faith in a team is if they...any time they sort of acknowledge the risks of their project or their code or their system, if they've taken the time to... especially if they've taken the time to document and share where the bad things are that could happen, that shows me that they not only have awareness around their code base, they also have awareness that bad things could happen, which is something that is surprisingly missing in this space, and it also shows that they've taken the time to write it down and that provides an additional level of accountability.

So all of these sort of tools, there's not one thing that's going to make a project trustworthy. There's not one thing that's going to make a project secure but if you take them altogether, a team that has a better chance of success is a team that has documents, they've written tests, they have a specification. They're engaged with the community for a long time. They're open to questions. They're open to answering the questions. They're aware that not everything is perfect and glorious all the time and that bad things can and probably will happen. And I'll say I think the first conversation I ever had with Robert from Compound I was very skeptical, and I was like so you're just going to have all this money on this smart contract? And how are you ever going to know it's secure? And he literally just responded and he was like well there's always a non-zero risk. There's never going to be a moment where I can go to sleep and but like everything's perfect, nothing bad will happen, and it really knocked me off my feet because I had been talking to so many people in this space where the answer would have been, oh, well, we had two audits by two different auditors and then we had it formally verified and we have 100 percent test coverage.

But it's actually Robert that gives me more faith in his team, that code, the compound protocol because I know that today and tomorrow and the next day that culture is going to always be on the lookout, whether that's the lookout for other hacks that may also affect the compound system,

success than even someone that has had all of the audits and used all of the tools.

Laura Shin:

Right. Yeah. That makes sense, and I love it that his honesty is actually what gave you confidence. All right. Well, this has been a fantastic conversation. I've really enjoyed it. Thank you, both, for coming on Unchained.

Dan Guido:

Thanks a lot.

Taylor Monahan:

Thank you so much, Laura.

Dan Guido:

Happy to be here.

Laura Shin:

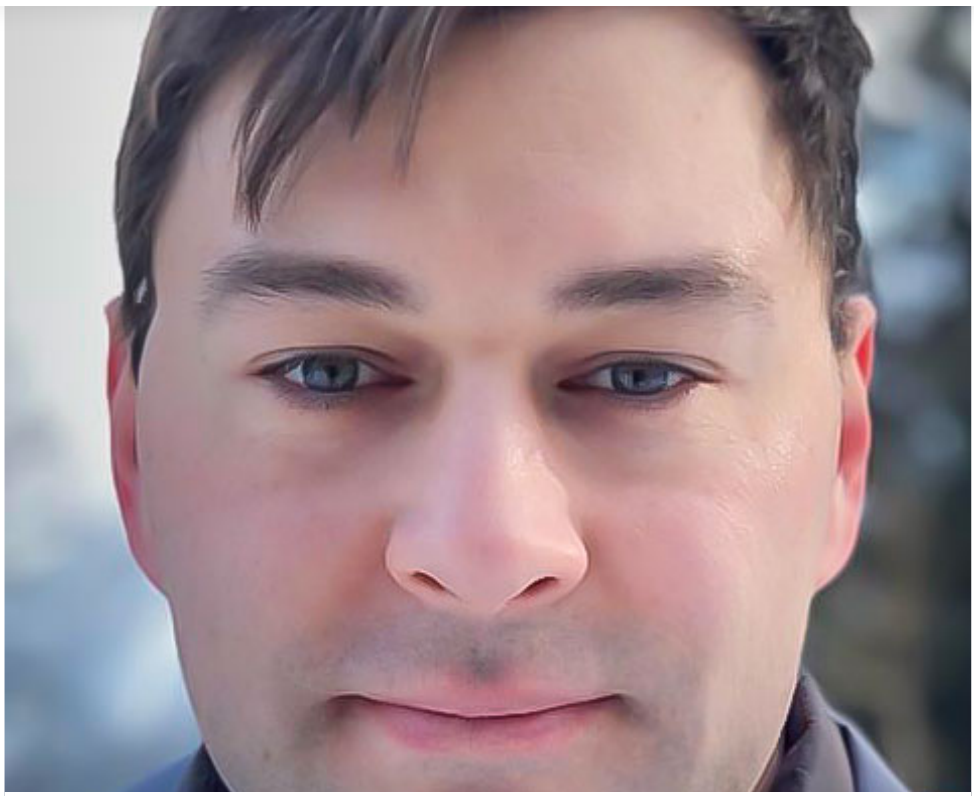
Thanks for tuning in. To learn more about Dan, Taylor, and DeFi security, be sure to check out the links in the show notes of your podcast player. Whatever your favorite crypto meme is, Lambos, unicorns, or the guy fox mask, it's probably on the Unchained rabbit hole t-shirt. Check it out at Shop.Unchainedpodcast.com and also be sure to check out our hats, mugs, and stickers, too.

Unchained is produced by me, Laura Shin, with help from Fractal Recording, Anthony Yoon, Daniel Nuss, Josh Durham, and the team at CLK Transcription. Thanks for listening.

Posted in: 2020 , Unchained

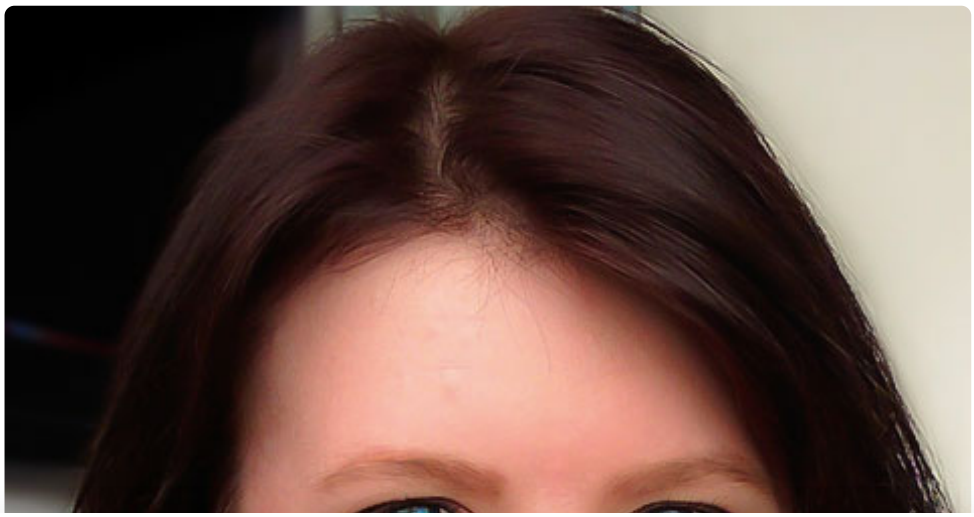
Tagged in: Audit , bitcoin , blockchain , crypto , Cryptocurrencies , cryptocurrency , Defi , ethereum , hack , investment

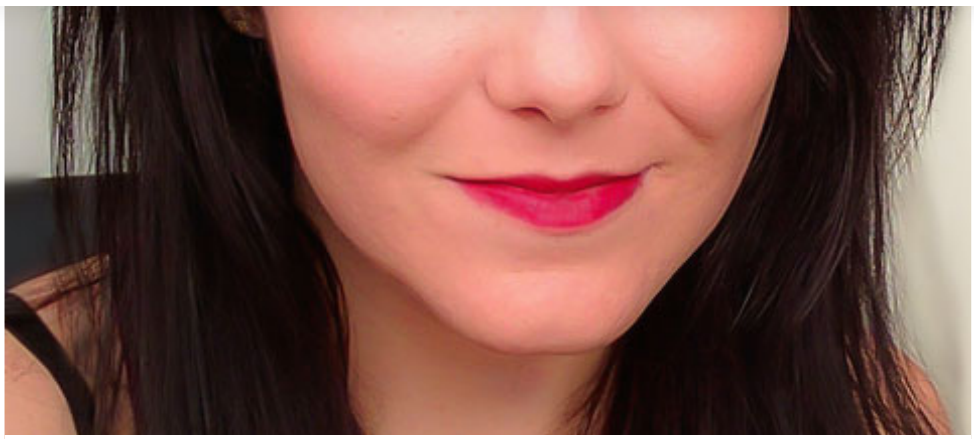
OUR GUESTS ON THIS EPISODE:



Dan Guido

Cofounder and CEO of Trail of Bits





Taylor Monahan

Founder and CEO of MyCrypto

Get the Smart Take

Sign up for Laura's newsletter on the top crypto stories of the week plus a preview of exclusive content!

Name

Email

JOIN FOR FREE

- Unchained
  - 2020
  - 2019
  - 2018
  - 2017

- Translated Transcripts
- Unconfirmed
  - 2020
  - 2019
  - 2018
- Newsletters
- Shop
- Resources
- About
- Contact
  - Advertisers
  - Donations
  - Tips & Pitches