Editor: Gary McGraw, gem@synopsys.com

Silver Bullet Talks with Nick Weaver

Gary McGraw | Synopsys

Hear the full podcast and find show links, notes, and an online discussion at www.synopsys.com/silverbullet.

Nicholas Weaver is a staff researcher at the University of California, Berkeley's International Computer Science Institute (ICSI). He also teaches courses at Berkeley. Weaver joined ICSI in 2003 as a post-doc after earning a PhD in computer science from Berkeley. His research focuses on network security, worms, botnets, and other Internet-scale attacks. He also works on network measurement.

ICSI is a nonprofit computer science research center. How is it funded?

It's almost entirely grant funded. As a researcher at ICSI, I'm very project and grant focused, and this is why I am doing more lecturing at Berkeley, because as a lecturer, I don't need to worry about research grants.

What are your views on ICSI tech transfer into the world?

As a research lab, we like building things that work. For example, the Bro Network Security Monitor was developed at ICSI, and that's being commercialized right now. Ten years ago, there was the extensible open router project, and there was a significant attempt to tech transfer that.

There are also systems that we've ended up building that have monetization models that don't match industry, but are productized. The Netalyzr network analysis tool that we originally wrote in Java in the web browser now runs on Android phones. We keep that running because it pays us in research results. We are able to turn the service into publications, and therefore we have a monetization strategy. It couldn't actually work out in the real world, but works for us. And we end up supporting a large number of users that way.

That's good stuff. You and I seem to share the same skeptical stance when it comes to cryptocurrencies and blockchain. Can you briefly give us a synopsis of your recent Burn It with Fire webinar?

I've come to this after five-plus years of watching the field and occasionally publishing on it. What it comes down to is there's actually three totally separate concepts. There is the concept of the cryptocurrencies themselves. There is the concept of the public blockchains, and then there is the concept of the private or permissions blockchains. Now let's start with the latter.

What is a private or permissions blockchain? Simply an append-only data structure with a limited number of authorized writers: aka, a git archive. There is nothing fundamental in a private blockchain that hasn't been understood in the field for 20-plus years. It's just it has a buzzword that causes idiots to throw money at the problem. If you see a private or permissions blockchain project, it means either one of two things. Either it's a delusional piece of techno-utopianism, or somebody smart in IT knows that there are real problems with what data you store, or how you access it, data provenance, and all this other stuff, and has bandied around this buzzword because idiots up in management will now throw money at this person to solve the real, interesting, hard problem.

That's one of the three. What about the other two?

The public blockchains are a global data structure where the idea is there is no centralized point of trust, but anybody can append to it. Now these systems are, let's say, not actually distributed as advertised. The Bitcoin blockchain is actually effectively controlled by only three entities, but in an attempt to be distributed, there is this religious notion that distributed trust is somehow good in and of itself. The result is systems that are either grossly inefficient or insecure.

The biggest tool that's used for these systems is what is called "proof

About Nick Weaver

Nick Weaver is a staff researcher at Berkeley's International Computer Science Institute (ICSI). He also teaches courses at Berkeley. Weaver joined ICSI in 2003 as a post-doc after earning a PhD in computer science from the University of California, Berkeley. His research focuses on network security, worms, botnets, and other Internet-scale attacks. He also works on network measurement. Weaver holds a BA in astrophysics and computer science. His thesis work was on FPGA architectures, but he's focused on computer security since 2001. He lives in Berkeley.

of work." And proof of work is best described as "proof of waste." The idea is that for somebody to rewrite the history, they have to do as much useless work as was done to create the history in the first place. Now this is great if you do a lot of useless work, except then it's inefficient. If you make the system efficient so you do not do a lot of useless work, you run into the problem of not actually having any real protection.

For example, Bitcoin, since the proof of work is paid for by the newly minted coins, ends up using as much power as New York City. It's just an obscene waste of energy. At the same time, these distributed public append-only ledgers only have been useful for cryptocurrencies. Now it's time to address the elephant in the room; the notion of the cryptocurrency itself.

Right? Back to one. Here we go.

Cryptocurrencies don't actually work as currency. They are provably inferior and can never be superior to the alternatives for real-world payments, unless you need what is known as "censorship resistance." If I want to transfer you $500 by PayPal, or Venmo, or whatever, we have these trusted intermediaries called banks, and they make it relatively cheap. However, there is a problem. If I want to transfer $500 to you for drugs or the like, these central authorities don't like it.

The only way to do censorship-resistant transactions without a cryptocurrency is cash, and cash requires physical proximity and math. One million in US dollars weighs 10 kilograms. That's a considerable amount of stuff to be lugging around. What a cryptocurrency is, well, let's do a direct to peer-to-peer payment system so that there are no central intermediaries, but let's do it electronically. This has been used quite practically for drug dealers, extortionists, fake hitmen, and all sorts of things like that. But if I want to do any payment that one of the central authorities will process, the cryptocurrencies provably don't work.

Let's say I want to buy a couch from Overstock.com using bitcoins. I have to turn my dollars into bitcoins, because I don't want to keep it in bitcoins because the price is jumping up and down. That is expensive. Transfer the bitcoin. That is relatively cheap right now, but it's been upwards of $30 in the past. And then the recipient on the other side has to convert the bitcoins back into dollars. You have these two mandatory currency conversion steps for any real-world transaction, and even Overstock, the one public company that supposedly embraces cryptocurrency, only keeps a few hundreds of thousands of dollars' worth of cryptocurrency, with the rest converted to dollars.

Cryptocurrencies do not work for legitimate purchases if you don't believe in the cryptocurrency. But let us suppose you believe in the vision of the great Satoshi. Then you don't want to use cryptocurrencies either, because they're baked in with these monetary policies that are designed to be deflationary. The first rule of a deflationary currency is never spend your deflationary currency.

There is one aspect of cryptocurrency that I think people don't understand, and it is this notion of tethers. Can you talk about that for a second?

There is a way to make a cryptocurrency work. You have to have an entity that takes dollars and gives you crypto dollars at par, and vice versa, that will take the crypto dollars and return you dollars. This is called a "bank," and these are called "banknotes," and it's recreating the 18th-century banking system. This can work, but one of three things has to happen. One option is you have regulation and enforce money-laundering laws and everything else, in which case you have a system that ends up being no cheaper or no more expensive than Visa, or Venmo, or anything else. What is the point?

Option number two is you have what is known as a "wildcat bank." This is a bank that prints banknotes that are actually unbacked. And this is a term from 18th-century banking.

The third option is a Liberty Reserve where you actually do back up your reserves. You redeem your digital banknotes, but you don't follow the money-laundering laws, in which case you end up being a guest of the federal government for the next 15 to 20 years.

At the same time, the money that the average person had is tied up temporarily or forever when the Feds shut down the institution. Tether is a specific cryptocurrency that promises to be backed

by dollars; they promise that there is this 1:1 ratio where you give them dollars, they give you tethers, and vice versa. The problem is this is almost certainly a wildcat bank because they managed to produce some 2 billion tethers in the space of a few months, and they are tied to a Bitcoin exchange that is otherwise cut off from banking. It may have been the direct reason why the Bitcoin price shot up so much.

Or they could be facilitating criminal money laundering, in which case those behind tether are liable to be guests of the federal government. This is, however, what actually enables most of the Bitcoin exchanges. Very few of the cryptocurrency exchanges actually are connected to the US banking system. You have Coinbase. You have Gemini, and you have Kraken (which should actually be shut down for other reasons of criminal activity, but that's neither here nor there). As for the rest of the exchanges, you can't actually transfer money into and out of them. These are where the hundreds and hundreds of different cryptocurrencies are actually traded on.

Tether has become this de facto reserve currency. If you look at Bitcoin trading volume, most of it is actually on tether-denominated exchanges and is not actually being exchanged for dollars, but these notional cryptodollars that may or may not be backed up, may or may not be a criminal enterprise—the flow just seems to continue on. It's really actually surprised me that it's lasted this long.

Yeah, it really is absolutely stunning this stuff. Thanks. That was extremely helpful. I think a lot of people need to have their eyes opened on this stuff, and you're one of the main people doing that.

I feel I have an obligation to. I kept looking at the field, and in the recent run up, I came to the conclusion that it's no longer harm-limited to a small population of self-selected believers. It is spilling out into the regular public.

Fortunately, I think the cryptocurrency space can die with proper application of regulation because of how the regulations already are, but it's become important for me to advocate for the need to clean up the space in that cryptocurrencies don't provide benefit to society. They don't provide benefit to all of us who aren't interested in committing crimes, but they do enable these problems. I think it is important to speak out. Another thing is the amount of scams in the space is just incredible.

Effectively every initial coin offering these days should be called a scam, because it is an unregistered security and wouldn't even pass the laugh test on Shark Tank. And we have got these people hyping smart contracts. Most of the cryptocurrency community seems intent on speed-running 500 years of economic history for choosing their bad ideas, but smart contracts are actually a new bad idea. The idea behind a smart contract is that I write a program that is not really a smart contract, it's a finance bot, because if it's a contract, you have this exception-handling mechanism called a judge in the legal system.

If I can walk up to a smart contract, say "Give me all your money," and it does, is that even theft? Well, it would be theft in the real world because we believe in justifying things, and this exception-handling mechanism of the judge and jury and all that. Smart contracts are instead—let's take the idea of a contract that is standardized and written in a formal way, it's called "legalese," and instead, rewrite it in a language that is uglier than JavaScript and has all sorts of pitfalls for programmers, eliminate the exception-handling mechanism, and then require that the code be bug free.

Except it's not bug free.

Oh, it's so amusingly not bug free. I like to use three examples. The first is the DAO, the Decentralized Autonomous Organization. The idea is, let's create a self-voting mutual fund for how we can invest our cryptocurrency in other projects. Now that there's actually nothing to invest was neither here nor there, but around 10 percent of all Ethereum at the time ended up in this basically self-creating, self-perpetuating, not-quite-a-Ponzi Ponzi scheme.

This was all fine and good until somebody noticed there was a reentrancy bug that allowed them to say, "Hey DAO, I am an investor. Give me all my money." And in the process repeat the thing as, "Hey DAO, give me all my money." And because there was a transfer then update, and you could re-entrantly call this code, it basically sucked all the money out.

The problem is, well, the money that was stolen mostly belonged to the people who came up with Ethereum in the first place. They basically did a code release that changed it and undid history. Their notion that code is law and there is no central authorities and no way to undo things was revealed to be a transparent lie when it's their money on the line.

Exactly.

So that's number one. Number two is the Proof of Weak Hands explicit Ponzi Scheme. Version 1.0 collected several million bucks before one bug locked it up so nobody could transfer any more money into it, and another bug allowed somebody to steal all the money in it. I think they're up to 3.0 now, which has yet to have a fatal bug, but we'll see how long that lasts.

Finally there is the Parity multisig wallet. One of the problems of cryptocurrencies is you can't actually store your cryptocurrency on an Internet-connected computer because if somebody gets onto your computer,

This article originally appeared in
IEEE Security & Privacy, *vol. 16, no. 4, 2018.*

they get your private key and steal all your money. We actually had this happen to us in the early days of Bitcoin, and if security researchers can't use Bitcoin on an Internet-connected computer, nobody can. The idea is, let's make it a two-party check system. We will have three private keys, and you have to use two of them to transfer the currency.

This gives you good controls if you can theoretically maintain at least two of your cryptographic keys. Some systems, like Bitcoin, offer it as a primitive. For Ethereum, it was built as a smart contract on top of things. This was the Parity multisig wallet, which collected some hundreds of millions of dollars, including an ICO by the guy behind the Parity multisig wallet. Until somebody noticed that there was a bug where you could go up to one of these wallets say, "Hey, wallet. You belong to me. Hey, wallet. Give me all your money," and started cleaning these out. And the only reason this wasn't a $150 million theft is somebody else noticed that this was going on, stole all the money first, and then gave it back to the victim once the victim had upgraded code.

Unbelievable.

Which gets better. Now there's the upgraded wallet code. For efficiency, everybody refers to the same wallet contract, and there was a bug in this contract. Some random loser came along and said, "Hey contract. You belong to me now," and the contract said, "Okey-doke. Yeah, I do." Okay, oh crap. This shouldn't have happened. "Hey, contract. Kill yourself." The contract committed suicide, and now $150 million worth of cryptocurrency is locked up and effectively inaccessible unless the central authorities, that aren't supposed to exist, change the code to unlock this. We're not done yet. The pièce de résistance.

The lead programmer and shining light behind this fiasco is the guy who invented the programming language in the first place. The problem is these things are designed to be non-upgradeable, but there are hacks that allow you to update them. If your money is tied up in somebody else's contract because their contract is the service, you have a choice. Either that contract has to have been bug free when created, not good, or that contract has to be upgradeable, in which case you have to trust that they upgrade the contract properly and don't cause damage or work against you in the process.

You have a central authority again.

You have a central authority. For example, there was a bug discovered in some of these smart contracts that run these ICOs, where somebody was able to create, what was it, 200 billion new tokens? Well, the people in charge of that particular smart contract were able to undo the process, but that means also if they can destroy the hack-created tokens, if you're invested in them, they can destroy your tokens too if they feel like it.

You have to trust them.

This is the ultimate irony in all these systems—their belief in this mantra that lack of trust and decentralization are good in and of themselves, ignoring the huge advantages you get with just even the slightest of smattering of centralized trust. Yet they end up building systems that aren't even decentralized. They build things that are orders of magnitude less efficient than they could be, but which have central authorities and aren't distributed anyway.

I think the real design decision was, "I would like to have all the trust belong to me."

No, the cryptocurrency community truly believes in this idea of decentralization; that you should have to trust nobody.

They're just bad at implementing it.

They don't understand the costs involved in that, and they cannot seem to ever implement it that way anyway.

All right, so onto a very personal issue. You suffer from depression that's treated by therapy and medication, and you talk about that so others can benefit from the good aspects of treatment and therapy. Tell us a little bit about that.

I've basically had in my life multiple depression meltdowns, and therapy and drugs saved my life twice as a student. And both times, after about a year, I'd just go off the medication, and a couple of years later the same thing would happen again. Just after the third incident, I realized that I didn't want to repeat that mistake.

So, when I'm teaching students, every semester I include in my first slide deck, the notion that yes, I've been there. I've done that. This is not good. There is help available. Every semester at least one student has proven that it's been worthwhile and they'll come up to me afterward.

Super important work. Last question, what is your favorite fiction book or your favorite fiction book you're reading at the moment?

Let's just say I'm a huge fan of *The Laundry Files*.

The Silver Bullet Podcast with Gary McGraw is cosponsored by Synopsys and this magazine and is syndicated by SearchSecurity. ■

Gary McGraw is vice president of security technology at Synopsys. He's the author of *Software Security: Building Security In* (Addison-Wesley 2006) and eight other books. McGraw received a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him via garymcgraw.com.