BitCoin CryptoCurrency

Description: This week, after catching up with a busy "Patch Tuesday," Steve and Tom explore the fascinating crypto technology developed to create "BitCoin," the Internet's decentralized peer-to-peer completely private online currency exchange system.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-287.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-287-lq.mp3

TOM MERRITT: This is Security Now!, with Steve Gibson, Episode 287, recorded February 9, 2011: BitCoin CryptoCurrency.

It's time for Security Now!, the show you need to listen to if you want to be safe on the Internet. And joining us to help us figure out all of the confusing things that could happen to you to threaten your security is the man who brought us ShieldsUP!, SpinRite, GRC.com: Mr. Steve Gibson. Good to be back with you again this week.

Steve Gibson: Hey, Tom. It's great to be with you for our second out of three weeks while Leo is roaming around the globe somewhere. Is he in Asia? Is that where he is?

TOM: I believe he is with penguins. He's on his way Antarctica.

Steve: On a cruise; right?

TOM: Yeah, we actually got to use the words "out to sea," literally. We were talking about whether we could contact Leo. He's actually been very communicative. He did a meet-up in Argentina when they stopped in Argentina. And he's been Instagramming and Twittering. So almost feels like I'm on the vacation with him, except having to work.

Steve: Fans everywhere.

TOM: Yeah, I know. All right, we've got a really good show today. We're going to be talking a little bit about a virtual crypto currency?

Steve: Yeah. It's something that a cryptographer, a Japanese cryptographer created about two years ago called BitCoin. And I learned about it, I think someone sent me a tweet about it, saying, hey, Steve, check this out when you get a chance. And so I dutifully noted it, jotted it down, and had a chance over the last week to dig into it more deeply. And I'm really impressed by what they've done and by the fact that this thing

really looks like it's the first solution to the concept of a distributed, non-central server, no central clearinghouse. I mean, it's like it's currency, it's Internet currency, which can work and is working. And there's just lots to talk about. Lots of cool technology in there, which of course is our angle from the crypto side. So we're going to talk about that this week.

TOM: You know, I have always been fascinated with the idea of online currencies, even before Flooz came along and sort of turned it into a joke back in the dotcom days. That is, like, the worst example. I think that did more to set back online currency than anything else that's ever happened. So I'm glad to see that a serious effort is underway, and I can't wait to learn more about it. We've also got some security news about throttling and do-not-track and some good stuff like that coming up in our updates, as well. Let's get up into the security updates. We've got a busy Patch Tuesday today. Well, this week.

Steve: Exactly, this week. Of course, last Tuesday was February 1st. So Patch Tuesday was the earliest it is able to occur in a month, that being February 8th. And both Microsoft and Adobe showed up. Microsoft had a large update. They fixed 22 different flaws, five of which were rated critical, sort of across the board in their operating systems and IE and server platforms. Seven had been publicly reported, and 15 were privately reported. And the good news is the nasty one that we have been talking about the last couple weeks, that MHTML flaw...

TOM: Oh, the MIME one, the one we were explaining last week, yeah.

Steve: Exactly, the MIME HTML, where, if you were to archive a web page which was malicious and then view it, it could get you. But there was also a way that a website could supply an MHT format page. There was a bug in the way this MHTML was parsed, and I'm really pleased that was fixed. Now...

TOM: We weren't sure it wasn't going to be fixed, were we.

Steve: No, because it was so, I mean, it was a short period of time, just before this Patch Tuesday. So it's interesting because sometimes Microsoft really seems to, like, be asleep at the switch; and sometimes they just jump on it. So I think they probably really wanted to avoid an out-of-cycle patch. This thing was being exploited in the wild. It was a zero-day exploit that was first discovered when it was being used to attack people. So they had that little quick fix which we talked about last week. And the effect of applying that, our listeners may remember, is only to disable the scripting in MHT archives. Which I would argue you could very well just leave off.

TOM: I was going to say, is that a good idea to have on anyway?

Steve: Exactly. It's like many things which default on, and which if we knew better we would just always have them off. And of course, if people had them, they would have never been potentially vulnerable to this problem. So I would be inclined just to leave it the way it is, if you did disable it. In any event, you'll have to do a reboot because this thing is actually part of Windows proper more than IE, although IE was the vector for exploitation. So that got fixed.

Also there was a longstanding and kind of well-known flaw in Microsoft's Internet Server, IIS, that is, for the FTP service. There was a way that a maliciously formed, deliberately malformed FTP command could gain bad guys access. Microsoft's defense was, oh, well, FTP service is not installed by default. But it's like, okay, fine. For Vista and Windows 7

users, if you had IIS loaded and were using an FTP server, as people who have some reason to do so would, there was a vulnerability there. That they fixed.

Now, the big news, though, is Microsoft did something that they almost never do. There was a non-security behavior change that they also released this last Tuesday. Two years ago we talked about - it was almost exactly two years ago. It was February 24th of 2009. On this podcast we talked about their bug fix for the broken disabling of autorun. That is, it's very well known that all kinds of worms, I mean, Conficker, for example, is famous for this, will jump onto, for example, USB thumb drives and use the fact that, when you stick it into a computer, it looks for autorun.inf and then runs in order to spread. So it used to be that you could, and very smart people would, disable that when they were setting up Windows the first time because it was...

TOM: It was actually one of the first web articles I ever edited for The Screen Savers in 1999 was Kate Botello telling folks how to disable autorun in Windows 98.

Steve: Right. And it's been a longstanding problem. Now, what's really funny - well, funny in a strange way, ironic, I guess - is that just this recent ShmooCon 2011, Jon Larimer from IBM's X-Force Security Division gave a presentation - it was a very early presentation in the morning, he thanked people for getting up so early - on how unfortunately Linux's desktops have been evolving to be easier to use and becoming more like Windows, and that as a consequence Ubuntu is now exploitable due to its support for autorun, which has just recently been added to the Linux desktop; whereas Microsoft has been burned by it so much, they're moving away from it. So...

TOM: Now, that is an oddity. I mean, it is the classic tension between ease of use and security, though, right there playing out in real life.

Steve: Well, and that's why Microsoft so infrequently takes away anything that they have previously had. They are so reticent to remove functionality. But this autorun problem with USB is a persistent problem. So what happened was, in February two years ago they fixed a bug where it wasn't actually disabled the way we thought it was. And then they came back in August, August 25th of '09, and for the first time they created an optional security update that would, for people who wanted to run it manually, would essentially do this for them. It would stop autoplay functionality on USB, on external hard drives, and on network shares, all of which were being exploited for various purposes.

The big news for this Tuesday is that they rolled it out formally as part of their normal Windows update process. Basically, it's installed non-optionally, and it turns off Autoplay for USB devices. So that's huge. What that'll mean is, I mean, the downside is, people who have been dependent on that will find that something that they're used to no longer works. They'll have to manually run setup or install or autorun.inf, whatever it is, however they normally would be starting something that's on a removable device that they plug in, as opposed to it happening automatically. Now, there are many USB devices which emulate a CD. And those will continue functioning.

TOM: Oh, okay. So this is not turning off all autorun, which is what I've seen a lot of people interpreting this as. It's turning off autorun for certain configurations, if you will.

Steve: Precisely. CDs and DVDs will still autoplay as they did before. But USB sticks - unless they emulate a CD, in which case Windows thinks it's a CD and will autoplay it - unless it emulates a CD, then Windows is just, from this point on, saying no, we just can't take the risk. Users are going to have to run this stuff manually because...

TOM: So there's no way to turn it back on. It's just disabled.

Steve: You know, I didn't pursue that. I'm sure you can. I'll bet you could go back into the registry and manually reenable...

TOM: Flip the switch, yeah, okay.

Steve: ...reenable autoplay. I'm sure you could turn it back on. But so what Microsoft is saying is, if you haven't manually disabled it yet, we'll disable it for you. If you want to come back later and turn it on, fine. Then we're assuming you know what you're doing, and you're going to ask for the behavior that you get.

TOM: Dr. Mom in the chatroom has a good point. Does that mean something like U3 still autoruns?

Steve: Precisely. U3 was what I was thinking of when I talked about a device which does emulate a CD because it shows you a CD, and it looks like that to the OS. So something like U3, you don't lose the functionality there. Which is kind of a nice compromise.

TOM: I guess it is, but how secure does this make us if somebody can just create their malware to emulate a CD?

Steve: Correct. You're right. That'll be the next thing is that we'll now move there. Anyone who's running Linux, who's concerned about what IBM's X-Force guy showed at ShmooCon, if you just Google "ShmooCon 2011" and then "Autorun attacks against Linux" - probably you could just do "Autorun attacks against Linux." But that will bring up - that's the title of this 51-minute YouTube presentation of this guy's talk.

And the point he's made is that in Linux, as in other operating systems, but specifically targeting Linux for his presentation, when you stick a USB device into a contemporary Linux desktop, all kinds of different levels of driver are engaged in order to connect with and recognize and mount the drive into the file system. And many of those devices, he contends, have not nearly been examined for exploitability as much as we would like. And he demonstrates taking over a Linux desktop that is normal default-configured, just by sticking in a maliciously formatted USB device. So again, Linux desktop users may want to check out that presentation. It was a good one.

TOM: We also got a big, thick stack of security vulnerability fixes for Adobe Reader and Acrobat.

Steve: Yep, they're catching up with - they had 29, 29...

TOM: Hey, they beat Microsoft.

Steve: They did, 29 critical security vulnerabilities which they addressed in the release version of Reader X, which you know they use an "X" for that, so Reader X, Reader 10.0. And also many of the same things were in Reader 9.4.1. So they're encouraging everyone to update to the latest version 10 of Reader and Acrobat, and that's 10.0.1. And then in their release notes they note that it also includes updates to Flash Player, keeping it current.

And just I needed to mention this because I guess there must be some people somewhere who are still using RealPlayer...

TOM: We could probably count them on one hand.

Steve: And that's the good news. Real, as we've talked about in the past, just was a horrendous security and sort of over-marketing exploitation approach to media players back in the beginning, really before Microsoft got into the media player business and sort of pushed them aside. There are still, I think mostly within corporate America, or in general, corporate Earth, companies that are standardized on RealPlayer. If you're using the .rm, .rmi, the standard RealMedia formats, in this case you're okay because this is specifically an AVI vulnerability. Quoting from one of the sites that was tracking this, they said: "A buffer is allocated according to [a] user supplied length value. User supplied data is then copied into the allocated buffer, without verifying [its] length, allowing the data to be written past the bounds of the previously allocated buffer."

I mean, this is classic buffer overflow attack. You ask the media, oh, how large is the data you're going to give me? And the media says, oh, let's call it 200 bytes. And then it says, okay, fine. Let me have it. And of course the media loads 5K and blows the buffer of 200 bytes that was allocated, and then stomps over the stack, and has just loaded executable code, which the system then runs when it tries to come back from the subroutine that was loading this.

So anyway, this is a classic buffer overrun exploit. It's in the current release of RealPlayer, only affecting AVI files. So if you are a RealPlayer user, you probably know it. Go over to Real and bring yourself up to date because the way this would be exploited would be just going to a web page that happened to invoke an AVI file in RealPlayer under the hope that you might have it installed. If you did, you could get taken over. So you don't want that to happen.

TOM: I might also add, if you're a Real user, go to Videoland.org/vlc.

Steve: And stop being a RealPlayer user, yes. Very good idea.

TOM: All right. Let's get into some security news. Firefox yesterday added something to its latest beta of v4, the do-not-track option we were talking about last week.

Steve: Yep, we talked about their intention to do so. I just wanted to let people know that it had appeared in the UI of Beta 11. It is not enabled by default at this point. Of course, it's not supported by default, by any advertisers that we know of in the world. But it's one of those chicken-and-egg things. The advertisers won't support it until the browsers ask for it. So I'm glad to say that Firefox 4 is asking for it. And I know that, as soon as I start using 4, I'll go there to the Advanced tab and say, yes, turn on do-not-track, and begin to get some experience with how that works.

TOM: I like that the option says "Tell sites I do not wish to be tracked." But it doesn't promise they won't because that's accurate to what it's doing. It's putting up a flag, but sites don't necessarily have to honor that flag.

Steve: Yeah, and, I mean, I can vouch for the pervasiveness of Firefox use. I mean, I know that GRC is going to tend to have a savvier user base come by. But by far the bulk of visitors at GRC are using Firefox over IE, which of course in our case is in the No. 2 position. But so that says that it's not as if we all have to sit around now waiting for Microsoft to do something before anyone's going to take this seriously. I just hear people more and more talking about that they're using Firefox. And of course Chrome is coming on very strong, too. Google, as we also discussed last week, has made some motion in this direction, this whole do-not-track deal. So the good news is, this has been a problem

for years, and we're beginning to see some solutions.

TOM: Hopefully we'll get to a standard. It's good that the different browsers are trying different things. Maybe we can see what works, what catches on. What I always hate is, there's a point where you could agree that, okay, that's the thing that works best, let's all standardize on that. Rarely does that happen. Usually we go through a long march of everybody sticking to whatever it was the started with. So...

Steve: Well, which we already have, for example, with NoScript that has its own format of do-not-track, different from what the Mozilla folks adopted, unfortunately.

TOM: Right, exactly. Within the same browser, even.

Steve: Within the same browser. So, like, Giorgio, when Mozilla announced this, Giorgio, the author of NoScript, he posted immediately, said, uh, you know, I already put this in here. Happy to have you guys use the same header. But why not use the same header instead of use a different header? So now the query that has - a query from Firefox of v4 Beta 11 that has the Mozilla do-not-track turned on, and is using NoScript with Giorgio's options turned on, will have multiple headers saying the same thing in different ways.

TOM: And nobody listening.

Steve: Exactly. And nobody listening at this point, exactly.

TOM: All right. Verizon is coming out with their own version of the iPhone this week. And they have very quietly announced some new policies regarding throttling the top 5 percent of data users, as well as some, what they're calling "content optimization."

Steve: Yeah, which I thought - and I wanted to mention this just because I thought it was - the details of content optimization I thought was really interesting. They said on a PDF that they made available on their site, quoting first this issue of bandwidth throttling - just I wanted to bring it to our listeners' attention for any of those who would be affected.

They said: "Verizon Wireless strives to provide customers the best experience when using our network, a shared resource among tens of millions of customers. To help achieve this, if you use an extraordinary amount of data and [thus] fall within the top 5 percent of Verizon Wireless data users, we may reduce your data throughput speeds periodically for the remainder of your then current and immediately following billing cycle to ensure high-quality network performance for other users at locations and times of peak demand. Our proactive management of the Verizon Wireless network is designed to ensure that the remaining 95 percent of data customers aren't negatively affected by the inordinate data consumption of just a few users."

TOM: I think, you know, this is a replacement for maintaining your network at proper capacity. They're worried that they're going to get some bad press if their network gets clogged. And so what's an easy way to do it? Throttle down some people. But if you want to do that, you've got to put a policy in place that explains who you're going to throttle down. So this doesn't - a lot of people are saying, oh, if you're in the top 5 percent you'll be throttled for two months. That's not exactly what they're saying here. They're saying, we reserve the right to periodically throttle you, basically when we need to.

Steve: Right, I agree. I think that, exactly as you said, they want to be preemptive. They want to say, look, just to let you know, if you are, I mean, really hogging

bandwidth. Because I got my Verizon iPhone yesterday, and I've got unlimited bandwidth use on it. That was the plan I chose. I know, I have a BlackBerry; now I have the iPhone 4. And I'm never going to be a heavy user. But I know that there are people who, I mean, they're sitting there watching all of their video consumption through all of the various online services now, and over time using a huge amount of bandwidth. So Verizon is saying, look, for people who are really at the top tier, as you said, we may need to throttle you.

Now, what's also interesting is, from a technology standpoint, I got a kick out of what they've acknowledged they're doing. And anyone who's interested in the details, I'm going to run through them. But you can see the whole document at VerizonWireless.com/vzwoptimization. So VerizonWireless.com/vzwoptimization. They said:

"We are implementing optimization and transcoding technologies in our network to transmit data files in a more efficient manner to allow available network capacity to benefit the greatest number of users. These techniques include caching less data, using less capacity, and sizing the video more appropriately for the device. The optimization process is agnostic to the content itself and to the website that provides it." So they're making clear they're not wanting to go upset all the Net Neutrality people. They say:

"While we invest much effort to avoid changing text, image, and video files in the compression process, and while any change to the file is likely to be indiscernible, the optimization process may minimally impact the appearance of the file as displayed on your device. For a further, more detailed explanation of these techniques, please visit www.verizonwireless.com/vzwoptimization."

Now, I did go there and look. And I saw a couple things that I wanted to bring to our listeners' attention. First of all, this only applies over port 80, which is to say, HTTP. They have, as we know, they have no visibility into HTTPS, into SSL connections.

TOM: Well, that's a nice little workaround.

Steve: So, exactly, that is.

TOM: If you're with a website that honors HTTPS, of course.

Steve: Right. And the reason this is interesting is that they really are - so what they're trying to do is they're trying to conserve the air bandwidth, that is, bandwidth in the air. So if you were to go to a website, for example, that had a really large, low-compression JPEG - anyone who's actually ever made a JPEG probably knows that the compression that you set on a JPEG is variable. You can choose lower compression, higher quality, where the image stays, like, ultra crisp sharp. Or you can make a JPEG image, the file, physically much smaller at the cost of some fuzziness. Basically, in terms of the type of compression JPEG uses, something called discrete cosine [transport] compression, DCT, it's expensive to transmit the data of a sharp edge. It's much less expensive to transmit the data of a gradual change, the way this type of compression works. So if you back off from requiring your images to have sharp edges, then you can get a much greater level of compression.

So what Verizon is doing is they're literally parsing the stream, looking at the objects which are being downloaded from web servers, and here they're saying they're reserving the option to change the data. They will take a low-compression JPEG and recompress it to a higher level in order to minimize its size. They will even transcode video, on the fly,

across formats. They'll go from, for example, they might take an AVI that's low compression, or RealMedia or something. If they know what your device is capable of, they will transcode it, and this document talks about this, to H.264, which as we know is a much higher quality compression for bitrate. And so what they're saying is, at their discretion, they're going to preserve the bandwidth of their over-the-air service and compress things.

Now, what's really amazing is that they're not doing it based on URL or even filename. They look at the first 8K, which is typically multiple frames of a video, to determine if they've seen it again. So they're watching the start of your video and using that to key their own caching technology to see whether they have already seen this video before and compressed it for somebody else. And, if so, they switch you to that stream, and that's what they send.

TOM: So you're sharing streams.

Steve: It's aggressive. I mean, this is aggressive optimization. Maybe they weren't carrying the iPhone until now because they weren't ready for it.

TOM: That very well may be true with all of this work. And couldn't they have - this is a cheap shot, but I'm going to say it anyway. Couldn't they have spent that time and money on capacity?

Steve: Well, this is a long-term investment. I salute them for doing this. And this is some serious technology. I mean, this is state-of-the-art caching and WiFi bandwidth optimization. It'll be interesting to see if users notice any effect. I mean, you could imagine, that, like, you could have two videos that start the same because they were edited from the same source material, but then are different. And their cache could be fooled by that.

TOM: I was going to ask about that. I wonder when we get the first people on purpose spoofing videos that are popular to deliver some maybe images that people weren't expecting.

Steve: Yeah. The other thing that they're doing along the same lines is that they're deliberately sending only enough video ahead to keep your player running. That is...

TOM: Yeah, I was thinking this would be a nifty way to take advantage of their transcoding, if you wanted to change videos to H.264. But you don't actually get the whole file.

Steve: Yeah. And again, they're being smart about this. They're recognizing that many people don't watch the whole video that they download, yet they downloaded it all. So Verizon is saying, we're going to be buffering in your player, but we're only going to stay enough ahead that, if you stop watching something after a few minutes of a 51-minute presentation, for example that YouTube I just talked about, then we won't have wasted our over-air bandwidth delivering video that was never seen. So potentially this is all good, as long as it doesn't cause problems. I would say it's tricky technology. I salute them for being this aggressive. I hope it doesn't have any downside. I imagine there will be people who'll be playing with it.

TOM: I think you're absolutely right about that. The other thing I've been seeing in the news lately are a lot of reports about how mobile is now the new battlefield for malware because we just had a report yesterday saying that smartphones outsold PCs in the last

quarter of 2010. So there's some news from McAfee about this?

Steve: Yeah, Symantec had issued a report. We're beginning to see reports from the major security guys, and McAfee just, I think it was yesterday, issued their report where - and paraphrasing them, they didn't use the phrase "new low-hanging fruit," but that's how I would describe it. What's happening is that PC technology, and Windows specifically because it's been such a target for attack - I mean, what, this podcast is in its sixth year. Leo and I have been talking about Windows security, Internet security, security, security, security, every single week for six years. Meanwhile, smartphones come along and are being adopted, as you just said, at a fantastic rate, and often, frankly, being used by people who are even less tech savvy than Windows users, who have figured out what it is they have to do in order to be safe.

TOM: Less of a barrier to entry, so to speak.

Steve: Yes. And maybe there's even more temptation. Maybe it's just that people aren't yet as afraid as they need to be about phones. But arguably, a smartphone, I mean, we know that it's running a full operating system now, given all that they're able to do. But the thing that malware wants more than anything else is connectivity. And while it's true that PCs are connected, I would argue smartphones are even more connected. I mean, there's more channels. You've got text, you've got all the social networking things, you've got email, you've got web browsing, and you've got applications, which, I mean, and this is of course a problem and a concern over on the Android platform, where people you don't necessarily know real well have created these things that look like, oh, wow, I really need that, and bang, now it's loaded in my smartphone. Well, what is it doing? It has all access to potentially this massive communication resource on the little computer that you're holding in your hand.

So I just wanted to say, once again, that we are seeing sort of the people who are watching security trends, they're saying that malware exploits are trending rapidly in the direction of smartphones. So for our listeners, just stay on your toes.

TOM: All right. We're going to get into our main topic, BitCoin, a digital currency. But I know you have a testimonial for SpinRite to read first.

Steve: Just, yeah, a nice letter that someone, a listener of ours named Mark Folkart, sent, with the subject "Yet another SpinRite story." He said, "Steve, been listening to your podcasts and following you for a while. I wanted to say thank you and relay yet another success story of SpinRite. I've been a computer consultant for over 10 years and had a client come to me," he says, "(CIO/Director of IS for medium-size foundation) with her husband's dead laptop. Her husband is not a client, but you know how that goes. He works for a large brokerage company I won't name. It had all his client/contact information on the laptop with no backup. He had gone to his IT department, and they were unable to assist him. At our urging, they unencrypted the drive and returned it to him still broken. And his sales database was still inaccessible and trapped locally. Couldn't even slave the drive. I used a copy," and he says, "(they had purchased a licensed copy) of SpinRite and went to work. Less than two hours later we were back in business. He had his contacts back and a working machine. Although I received no direct compensation, it certainly increased my credibility to a good customer, and how do you put a price on that? I will continue to use and recommend your product and just wanted to say thanks. Sincerely, Mark Folkart." So Mark, thank you for the great note.

TOM: It's amazing to me that an IT department wouldn't - and I've had it happen. I won't name the workplace, but I have been in a place where my drive crashed, and I was like, hey, can you recover the data off this? "Nah, can't be done," is what I was told. I

was like, well, no, it can. There are ways.

So our big topic today is BitCoin. You called this a "crypto currency."

Steve: Well, it's really, really clever. The reason that I sort of fell in love with this for the moment is, as I plowed in, I just got a big kick out of the way that the many problems associated with a sort of a floating currency, meaning a currency that isn't anchored by any central bank, there's no state sponsorship for it, I mean, and it's a real thing. Anyone who's interested, and I would encourage our listeners, if this podcast and what they hear about it makes them curious, go check it out. Just put "bitcoin" into Google, and you'll start seeing pages of stuff. And about two years ago the project was registered, a little over two years ago, by a Japanese cryptographer, Satoshi Nakamoto. And it's an open source project on SourceForge, so none of this is black art stuff.

The goal is to really solve, I mean, to offer an honest-to-god, non-hobby-level, but industrial-strength, Internet-based, peer-to-peer currency where real value can be exchanged between two parties without any intermediary being involved. And that's one of the trickiest things because you've got all kinds of problems. First of all, where does the currency come from? What creates the currency? How much currency is flowing through the system? How do you monitor that and regulate it? How do you prevent it from being inflated? How do you keep people from fraudulently creating currency? How do you keep someone from, if they have some, from reusing the same currency? All of that has been solved with this system in some very clever and very new ways. Which is really what captivated my attention on this. So there was...

TOM: So wait a minute. So we have currencies. We have euros and yen and dollars. How can you invent a currency? What makes that work?

Steve: Well, okay. So, think about it, a currency is nothing really but an agreement among the parties that this synthetic thing has value. Once upon a time, when the dollar was anchored to a gold standard, the idea was that there was gold backing up dollars. And so when you had a so-called "promissory note," it was equivalent to X amount of gold. And we were of course famously taken off of the gold standard. The problem was we needed more money than we had gold; so we had to disconnect, in the case of U.S. dollars, we had to disconnect U.S. dollars from gold because we literally needed to create more money than we had gold to back it up.

TOM: It's kind of that incredible innovation in human society, when you think about it, that this works at all. Because it started out you would carry around your chickens because you just wanted to trade what you had of value for what the blacksmith had. That got inconvenient, so gold became a good standard because everybody valued gold, and everybody kind of had the same value of gold. But we've gone from that to this sort of agreement that, well, I'm going to agree that a dollar's worth of work is worth a dollar's worth of merchandise, and it doesn't have to be backed by anything. We'll all agree that that's the way to pay stuff. So I guess that's all they have to do is get enough people to agree that this currency is valuable?

Steve: Correct. Well, and notice also that we chose gold because it was scarce. We didn't use water, for example, because you'd just go over to a stream and dip your bucket in. And the problem is, of course, anybody could go do that. So water...

TOM: There's a famous scene in one of the Douglas Adams novels where they decide leaves will be their currency. And it has the same problem.

Steve: Well, of course money grows on trees, so, yeah.

TOM: Right, exactly.

Steve: And so we chose gold because it was scarce. And famously in the days of individual gold miners, they'd go out and try to find it because they would - basically they were creating more currency to put into the system at a controlled rate. And initially, when there was lots of gold around, we were digging it up and turning it into bars and coins and so forth. And over time, it became increasingly difficult for us to find more gold, so it became increasingly scarce, and its value has increased. So...

TOM: And in some ways we have a virtual currency with the dollar and the euro and all of these. And in some ways that is a little more fair because someone can't just go out and find a bunch of money, unless they're robbing a bank, I guess. But, you know, you can't just go digging in the hills and luck into a bunch of money. It has to be earned in some manner.

Steve: Right. So what has been created with BitCoin has all of these attributes. There is this concept of bitcoins, the currency - in the same way that the abbreviation for U.S. dollars is USD, and euros is EUR, BitCoin's abbreviation is BTC, bitcoin, BTC. And so this network of computers exists now on the Internet, peer to peer. You can go to BitCoin.org and download a program for Windows, Mac, and Linux, which is open source, and install it on your computer, and tell it to start generating bitcoins. That is, literally start making money.

TOM: So you are making money out of nothing, just by being a member? I mean, how does this - this just sounds like some sort of BitTorrent situation.

Steve: I know. It sounds wacky, but...

TOM: Yeah, yeah.

Steve: So you are making money. The way you make money is by processing transactions within the bitcoin system. So, and this is complicated, but unfortunately it needs to be complicated in order to be robustly secure, which it really is. In the FAQ at BitCoin.org, in the FAQ there's a link to the original PDF that Satoshi wrote that describes in greater detail how this works. But the idea is that you want a transaction trail of every single transaction between two parties that has ever occurred. And they're occurring all the time.

Now, this is not just - this currency is virtual, but it has been anchored now to real currencies. There are websites that will trade real currencies for bitcoins. At this point in time, about two years after it was launched, the current currency trade of U.S. dollars for bitcoins is about 1:1. I think it's, like, 93 cents for a bitcoin. And there are organizations which accept bitcoin payments. The EFF, the Electronic Frontier Foundation, accepts donations in bitcoin currency. There are programmers who will work and accept payment in bitcoins. There's a, I think it's called Trade, a trade link at BitCoin.org that shows a page of lists of all the currency exchanges that exist now, and then a growing number of organizations and companies that will accept bitcoin currency as real. So I know I...

TOM: Okay, let's back up a little bit here. If I can just create, by running the program, money, aren't we running into the leaves and water problem, where we just get runaway inflation and the currency is valueless?

Steve: Yes, except that it's all controlled. The way it functions is that new coins, new bitcoins, are generated on the network when a node - and, for example, if you're running the program, you are one node - when a node finds the solution to a hard problem. Now, this is really very clever the way this works because it prevents people from being able to create currency at will.

Back in '97, I think it was, 1997, someone named Adam Back came up with a concept for antispam, which he called "proof of work." The idea was that spammers function because they're able to just spew out email at virtually zero cost. It doesn't cost them anything to send out email. So as a consequence we're all being deluged with email, which it's expensive for us to receive, not expensive for them to send. So Adam said, what if we come up with a way of making it expensive for someone to send email?

And the way we do that is, we create a computational burden which we don't have the technology to short-circuit, where they have to do a substantial amount of work in order to sort of validate an email. And on this show we've talked about hashing a lot. Hashing of course is a valuable technique that takes an arbitrary length input and turns it into, hashes it down into a so-called "digest" of a fixed length. So imagine, like, take SHA-256, which is the secure hashing algorithm which produces a 256-bit result. Imagine if, in order to qualify for sending email, you have to hash the email header such that some number of the first bits out of the 256 bits are all zero. So if you just hash an email header at random, the most significant bit has a 50 percent change of being a one or a zero. So you increment a sort of a fudge factor and then hash it again until you get that first bit that's a zero.

But say that to qualify the header has to have a hash where the first 20 bits, for example, are all zero. Well, it's going to take 2^20 operations to guarantee that. So an average, half that number of hashes have to be tried. So the idea is this forces someone to do a huge amount of work fudging the header in order to get all, like the first X number of bits of the hash to be zero. So in practice you could set the difficulty so that it might take somebody two seconds to do the work on a 1GHz PC. But that would mean that it takes a spammer two seconds per email, which is vastly more computation time than it takes them now.

TOM: And so on an individual basis you don't notice that that much.

Steve: Exactly.

TOM: But if you are trying to send vast amounts of email, which I guess could negatively impact legitimate bulk email like newsletters and things like that, too.

Steve: And actually that's exactly why the idea did not take off was that it was still - while, yes, it would be burdensome for spammers, exactly as you said, there are legitimate mass mailers. And if we did anything to allow them through, then the spammers would come through, too. So it had to be all or nothing. And it was too much work for legitimate mass mailers. But it was a really interesting concept. And Satoshi borrowed that concept that Adam Back proposed back in '97 for this.

So here's the way it works. So imagine that there are, among all these peers, there are people exchanging value. They're exchanging bitcoins. A bitcoin exchange is somebody wants to send somebody else some bitcoinage. So the whole system works with an asymmetric key system, a public key system where they have both a public key and a private key. They take some amount of bitcoinage and put their public key, sort of associate or include their public key in the transaction, also the public key of the person it is being sent to. And then they sign it with their private key.

So what that creates is, that creates a transaction that only they could have originated because they're the only ones who have their private key, which they keep secret. That transaction is broadcast into this peer-to-peer network, to all the nodes in the network, and everyone's transactions are broadcast. Now, it's easy for anyone to verify that transaction because they know the public key of the signer, and that allows them to verify the signature. They can't sign it themselves, but they can verify the signature. So that allows them to verify the transaction. Now what we have to do is we need to prevent that person, who's just depleted their bitcoinage by giving some away, from giving the same bitcoins away again. And so that's clearly one of the hard things to solve about this.

So the way we do this is, every so often, all of the transactions which have occurred since, okay, there's sort of a problem of chicken and egg here because I have to explain multiple things at once for this thing to hang together. There is this notion of blocks. A block is a collection of transactions which have been sort of adopted by the network. And the block, which is this collection of transactions, is the thing which work is done to create. In the same way that I was talking about work being done to create this special hash for email headers, the work being done to create this block is what all the nodes on the network are busy doing.

So all the nodes receive transactions. And a block is chained to all the previous blocks by taking the hash of the previous block as part of the next block. Which means that essentially you have a forward-moving chain of blocks which are linked by the hash of the previous block. There is a genesis, what's called the "genesis block," which was created on January 3rd of 2009. So just a little over two years ago, when the system began, there was an anchor block which is embedded into all of these nodes, into the code in the nodes. When someone downloads the program and turns it on, they go to an IRC chatroom, that is, the code autonomously goes to an IRC chatroom, joins the room, and that's how it learns about all the other nodes or many of the other nodes on the network. It then interconnects to them and receives the entire history of all previous blocks, that is, this block chain, anchored by the genesis block, all the way to the most recent block that anyone has created. So, and...

TOM: That sounds like it could become computationally extensive over time, though; right?

Steve: Yes, except that there's another clever thing. It turns out there's a way to compress these so that, once the blocks are old enough, and no one cares about the individual transaction details, then you no longer really need to care about them. The idea is you need the transaction details long enough to make sure that nobody - so that the transactions details are available in the network so that no one is able to reissue the same bitcoins again. But at some point then it becomes impossible for them to because the blocks become old enough. And you do not need to - it turns out you're able to compress these blocks and make them a lot smaller. So, and I think the growth rate is estimated at something like 4.2MB per year would be the maximum amount of storage that this architecture requires. So it ends up really not being very much over time.

So what happens is there is this sort of chain of blocks. Now, all the nodes in the network are competing with each other to create the next block. And it's the node which wins, the node which first does the amount of work required to essentially create the next block that earns 50 bitcoins. And this all sort of scales in the right way. I'll explain in a second.

So all of the nodes are cranking away. They are taking all the transactions which have not yet been encased in a block, and they hash all of that along with the hash of the

previous block, which that anchors them together and means that you're not able to create a block that isn't linked to the prior one, hash it all together, and then there's a certain amount of difficulty which is of finding a block that functions by exactly, as we were talking, having a hash with some number of zeroes from the left end going down. And at the moment, I think that number is 12 at this point in time. So all the nodes are tweaking a little fudge factor in the hash, trying to build a block which has 12 zeroes at the leading part of this 256-bit SHA-256 hash. As soon as the node finds it, it declares success, broadcasts that to the network.

Remember that, while it's extremely difficult to find the pattern that makes the hash, it's incredibly easy to verify it. Verifying the hash just requires doing the hash of the block and seeing that, oh, look, somebody did create a block that's got all those zeroes. And the first transaction in any block is paying yourself 50 bitcoins. But it's only if you can make that block valid that then that transaction in the block of paying yourself 50 bitcoins is validated by the network. So...

TOM: We've actually had two blocks created since we started explaining what blocks are, by the way. Their little estimate of how there are people out there using this.

Steve: Yes. In fact there's a site called, I think it's blocktrack...

TOM: BlockExplorer is the one that I'm at.

Steve: That's the one.

TOM: And that's how I'm keeping track of this.

Steve: Yep, BlockExplorer allows you, it's a website that is participating in this peer-to-peer network which allows you to go look at the history of all the blocks that have been created so far.

TOM: Now, is this a worry. that all of your finances are now going to be in public? Can people look at this and figure how much money you're spending and who you're giving it to?

Steve: Well, that's one of the other beauties is that the only thing which is known - this is a completely anonymous currency system. I mean, like more anonymous than anything else. The only thing that is known is your public key. So when you download this software and fire it up on your machine and start it running, the first thing it does is to create a key pair. And so you will see, for example, if you find the EFF bitcoin donation, they show their public key. And there are various other organizations that accept bitcoin. They show their public key.

So when you look at the history of transactions, all you're seeing is this random ASCII gibberish, which is the public key converted into ASCII. And people keep their private key private. But there's no way of knowing who is behind any public key. And the bitcoin client will happily produce key pairs till the cows come home. You can make more key pairs anytime you want. So you're not even - there's not even any way to track somebody by, like, oh, look, there's the same guy who did a transaction here. He did it here. Only if you did not create another public key would that be the case. But you are free to create new - essentially the public key is a temporary, pure binary representation of you, which you're free to retire and create a new one anytime you want.

TOM: I just downloaded it and started it. It's not creating - it says I'm not connected. It's

not creating anything. Is that because I'm not in idle time? It has to be idle to start generating those coins automatically?

Steve: Well, yes. And, okay. So many things have happened in the last two years. First of all, this began to get traction, and people began having fun with this. The way the system works is - and I need to get this right - is the coin creation rate is 300 coins per hour within the entire system. And your CPU speed, the ratio of your CPU speed over the total CPU speed within the entire bitcoin network, determines the probability that you will be able to solve the puzzle of creating one of these blocks. So it's estimated, for example, that at this point, I think it was December 2010, so about two months ago there were enough nodes actively cranking away that it would take you about a year to generate 50 bitcoins. That is, so you're not going to see it happen quickly.

Now, what happened is, as these things started getting valuable, started becoming worth something, I mean, you can actually trade, if you happen to get lucky, and your node solved the most recent block that everybody was working on before anybody else, you'd get 50 bitcoins. Today there's an exchange that will transfer that, in U.S. dollars, for example, into your PayPal account. So you actually can make money.

Now, you can imagine, then, that people said, wait a minute. This seems like a good idea. Well, there's something that's much more powerful than even multicore CPUs. And that's GPUs, graphics processing units. Now Google "bitcoin miner," as in a gold miner. What's happened is that there are people on the 'Net that have built bitcoin-creating boxes with as many graphics processing units as they can get, with fans cooling them, they're overclocked, they're pouring Freon over them. These things are running 24/7.

TOM: Like oil derricks. They're drilling for bitcoin.

Steve: They really are. They're literally creating bitcoinage. Now, the cool thing is, all of this was anticipated in the original system because the immediate response to the bitcoin network of the presence of massive bitcoin computation power, which essentially allowed the people who had these machines to be printing money, minting bitcoins with a much greater probability than somebody who just had a CPU running along, the system automatically changes and changed the problem difficulty in order to stabilize the rate at which coins are coming into the system.

And here's the deal. There will never, ever, ever be more than 21 million bitcoins created. The way this works is that the difficulty of this problem that is being solved, that is, this hashing problem where you're trying to find leading zeroes in the hash, it's adjusted continuously by the network. So that in the first four years of the bitcoin network, and we're two years in now, in the first four years half of that total number of bitcoins will be created, that is, 10,500,000 bitcoins will be created in the first four years. In the second four years, half again, that is, only 5,250,000 in years four through eight. In years eight through 12, that is the next four years, again that amount is halved. And so the rate of coinage creation will be decreasing exponentially, leveling off so that, in the far future, only 21 million will ever be created.

So we have a controlled and known rate of inflation within the system. And it makes sense because, initially, as the system is coming online, as goods and services are being made available and are trading within the system, you want to have more currency being pumped into the network so that you have bitcoins to trade. But you don't want it to go forever.

Now, the problem would be, of course, if we absolutely cap the total number of coinage at 21 million, and there comes a much greater demand for this, the tendency is to want

more. Well, the solution is that you're not forced to trade in integer amounts of bitcoins. That is, the UI right now gives you two decimal digits of coinage. So you're able to create, for example, you could exchange 0.01 of a bitcoin, but the technology supports eight decimal digits, although right now we're only using two. So that allows for deflation over time because we're absolutely capping the total number of coinage at this 21 million mark. And we know that it's going to be declining over time, and it doesn't matter how much GPU power is put into the system. The system adapts so that the problems being solved scale - the difficulty scales up to balance the amount of processing power in the entire network.

And some people have commented that the question then becomes, are you spending more money on electricity and cooling for these crazy bitcoin-generating engines than the money you produce? And over time it looks like that will be the limiting factor. It's like, yay, I've created this insane work machine to create bitcoins. But, gee, you know, my electric bill went up more than the money that I'm making.

TOM: And PG&E isn't talking bitcoins yet for my rates.

Steve: Exactly. So there's also a cool site that I got a kick out of called the BitCoin Faucet. The BitCoin Faucet...

TOM: I was just trying to use that. Their rate limit has been exceeded. I think everybody's going to get their - explain what it is.

Steve: It's just a fun way that somebody can get some free bitcoins. People who have them can donate them to the site, and the guy thanks you very much. And as long as he's got enough supply, he'll give you some bitcoins. At the moment, when I looked before the podcast, he was giving 0.05 bitcoins per visitor. When his balance of available bitcoins that he's able to distribute for free is high enough, he increases that. But if he falls below a certain mark, he decreases it in order to conserve his supply.

There's an online buyer and seller escrow service, so that two people are able to agree that they're each happy with the exchange of whatever it is they exchanged. For example, in the real world, in order to allow a bitcoin transaction to occur, there are a number of online exchanges where you're able to buy and sell bitcoins. There's online charts where you can look at the rate at which bitcoins are being bought and sold, and their relative currencies. This is available in a huge number of currencies and a whole bunch of languages.

And essentially it is extremely cool crypto which, I mean, this has been pounded on and looked at. And it looks to me like the guy has solved the problems and has created a virtual currency that floats all by itself, that is completely private, that, I mean, obviously you need somebody who's going to agree with you that you want to exchange this coinage. But this thing exists, and it's taking off, and I wanted our listeners to know about it. It's just very cool.

TOM: Really fascinating stuff. I've been playing with it while we've been talking, too. I have no balance, though. And the faucet is off right now. But at least I'm connected, finally. So I...

Steve: Oh, so you did find your network.

TOM: Yeah, it finally found the network. I have no connections yet, though. But it is connected.

Steve: One of the things they recommend, if our listeners want to play with this, is you will get much better connectivity if you do port forwarding of port 8333 through your router, which typically most people have, or your firewall, to the application. In that way, other nodes that are informed about you are able to make incoming connections to you, and it's not just you making outgoing connections to them. And that'll allow you to participate in the network. But I would encourage our listeners to poke around, as you have been, Tom. There are pictures of these people's GPU boxes that they've made, with all these fans all over them, and discussions about the giga hashes per second (GH/sec) in the network. I'm trying to think, I made some notes about it somewhere about the...

TOM: Oh, the average rate of block creation?

Steve: Oh, yeah, 186 GH/sec is the total network hashing strength. That is, there are 186 billion hash operations being performed within the entire network, trying to solve the problem of the next block creation. The one who does gets 50 bitcoins. Oh, and that number also decreases over time. So that it's, for the first 210,000 blocks, the value is 50 bitcoins per block, if your computer solves the puzzle before somebody else's. Then for the next 210 blocks, that's cut in half to 25 bitcoins. Then for the next 210,000 blocks to 12.5 bitcoins, and to 6.25 bitcoins, and so forth on down.

So this whole system is designed to scale correctly and basically create a secure, stable currency with real-world value, which it has now. I mean, you can buy and sell bitcoins. If you wanted to, you could take a hundred dollars and go buy some bitcoins. And they're electronic currency. You could then send those anonymously to someone else, and they could cash them in to their own currency or back into dollars or whatever they wanted. I mean, this exists now, and it looks like it's, like, bulletproof. And the PDF explains they've really thought through what bad guys can do. The only attack which is known on the system would involve somebody with massive computational power spoofing the chain because it's this chain of blocks which provides the integrity for the system. But the longer the chain gets and the more good nodes there are, the more impossible it becomes for anyone with massive computational power to spoof the chain.

So the other thing that has happened is there's this notion of pooling. Individuals have become a little disenchanted with the fact that they've got their quad-core i7 cranking away 24/7, and they haven't made any money yet. They like the idea of printing money. The fact is, over time this is not going to be feasible. That is, the way to get rich is not to print money. It's like gold miners, like during the gold rush. You ran out, and you hoped to go strike it rich and find a vein of gold and make money. Over time, that just became less and less feasible because the ground had been picked over, and there just wasn't gold to be found. Similarly, ultimately, this will be - it's when the coinage enters the currency and begins to flow, people will be using it as a store of value.

But anyway, what I was saying was that people who have been a little disenchanted, they're joining pools of users where they'll all be working together and pooling their CPU resources and then sharing the proceeds appropriately. So you may not get 50 bitcoins, but you may get 2.5 because you are one 2.5 out of 50th of the CPU resource in a pool that collectively solved a block. So it's a way for people to have some of the fun of creating coinage out of thin air by doing the work that makes the whole network go. And anyway, it is the work that has to be done, the difficulty of doing the work, that keeps bad guys from being able to spoof the system because there just isn't any way to shortcut this hashing of the blocks. It's just brute-force work.

And as more good, like GPU-based systems come online, that hugely raises the bar that bad guys would have to scale in order to spoof because now, as a consequence of sort of this phenomenon of using GPUs to create bitcoins, the network has scaled so the rate of

coin creation is still tracking exactly what it should. But the amount of processing time being required to make that happen has just gone through the roof.

TOM: There's bitcoins in them thar Internets. Or something.

Steve: Anyway, just really, really fascinating. And, I mean, it works. We have a state-free, crypto-secure, anonymous real currency now that exists.

TOM: Check it out at BitCoin.org. It's really fun to play around. Dot org, not dot arg.

Steve: Argh, matie.

TOM: Yeah. There's no piracy in the bitcoin world. Thank you, Steve, so much for explaining that. That was really fascinating. I can't wait to poke around with this a little more. I think I'm going to have to do the port forwarding that you were talking about to get it. Well, no, I've got one connection now. So it seems to be slowly - I guess that's what I'd say to new people, if you're looking at this. Be patient. Give it some time to connect. You might try the port-forwarding trick. What port was it, again?

Steve: 8333.

TOM: 8333.

Steve: All this is documented in the FAQ at BitCoin.org. And the other thing that happens is, when you connect, the first thing that happens is your node needs to download the history of prior blocks. So that'll take a while. But you'll see progress things and so forth. And there is, on the UI, there's an option to say "Start making bitcoins."

TOM: Yeah, start making money. All right, folks. Don't forget to visit GRC.com. Steve's got some excellent products up there. Of course we talked about SpinRite. I've used ShieldsUP! over and over throughout my years since you created it to make sure that there's not any ports open that I don't want opened and all that stuff. Also, it makes you feel really good when you are locked down because you have this great - I don't remember exactly how you phrase it. But you have this, like, "That's impressive. I couldn't see a single port." So check it out, GRC.com. Anything else you want to talk about before we head out of here, Steve?

Steve: Think we've got it covered. We'll do a Q&A next week. So I encourage, as always, our listeners to swing by GRC.com/feedback. And if you play with BitCoin, you have questions, I'd love to hear them, and maybe we'll answer them next week.

TOM: For free. We won't charge you any bitcoins to answer them.

Steve: No charge.

TOM: All right, thanks, Steve. Thanks, everybody, for watching. I'll be back one more week next week. Leo will be back in two weeks from his vacation. Thanks for watching Security Now!. We'll see you next time.

Steve: Thanks, Tom.