SECURITY NOW!

Transcript of Episode #289

Proxied Surfing

Description: After catching up with the week's security updates and other security-related news, Steve and Leo discuss the many modes of operation of "Proxied Web Surfing" which are used to bypass firewalls and Internet filters, aid free speech, and alter the contents of web pages retrieved from the Internet.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-289.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-289-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 289, recorded February 23rd, 2011: Proxied Surfing.

It's time for Security Now!, the show that protects you, your loved ones, and everyone else on the Internet from predators and viruses and bad guys, and at least explains how all this stuff works. Here he is, the man who does that, the man of the hour, Mr. Steve Gibson of GRC.com.

Steve Gibson: And in this case, Leo, maybe even foreign governments, or your own government that doesn't want you to know what's going on or talk to other people or your corporations or your schools or whatever. We're going to talk about something this week that, oddly enough, in our sixth year, we've never directly discussed, which is proxied surfing, the idea of connecting to the web through proxy servers for a number of different reasons.

It sort of came onto my radar when I was reading about what was going on, of course, in Egypt in the last couple weeks, with them disconnecting the Internet, and then reconnecting it, and how strong and important the whole Facebook and social networking had become for organizing groups. And I ran across a couple references to some proxy sites that offer their services, for example, in China, that allowed people to use them in order to bypass the Chinese restrictions that were being imposed. And I thought, that's obviously a great topic for the podcast. We've never really addressed all the different reasons you would use proxies, and exactly how the technology works. So that's our topic for this week, in addition, of course, to other stuff.

Leo: Very interesting. And what's been interesting in both Libya, and it happened in Egypt, and it happened in Tunisia, and I expect it will happen everywhere else where they try to restrict the Internet, is companies coming along, saying, well, you can always use dial-up. And while we don't, in the West, think of dial-up as a decent solution, it's plenty for tweeting and emailing and setting up Facebook pages. And obviously I think the time has - I'm very bullish. I think the time has come. I think the Internet is really a disruptive force in a lot of ways, and it's very exciting.

Steve: Well, it's clear that it is because for a nation to take itself off the 'Net, I mean, that does not come at zero cost to the nation. And so...

Leo: And it doesn't come with success, I mean, at least in the case of Egypt.

Steve: Right, right. And so given the fact that it would be so disruptive to a country's operation to disconnect it from the rest of the world, you'd have to imagine that the government that decides to do that is weighing the pros and cons and looking at the benefit they'll get from being repressive in that fashion, meaning that it is so powerful for their citizenry, in this case, that they want to disconnect from the rest of the world, to have access to it. So, yeah.

Leo: Well, let's - we're going to get to that subject in just a moment, of course. But before we do, how about some security updates?

Steve: Absolutely. It's been quiet because we had the second Tuesday of February was a biggie for Microsoft. Things have been busy. And welcome back, by the way. Tom did a great job for us.

Leo: Thank you for reminding me. Thank you, Tom, for filling in. I'm so sorry. Tom, I owe you an apology. Yes, three shows he did.

Steve: Yeah, he did a great job for us.

Leo: Good, thank you.

Steve: I remember being very nervous at the idea of not having you. Of course Tom stood in once before and did a great job. And this time it was like, oh, hi, Tom, welcome back. So in fact I had to send my email to someone else. I almost sent it to Tom this morning with all the show notes. I said, wait, wait a minute, Leo's back.

Leo: I'm back. I'm back, please. I had a great time. It was a wonderful trip. I'm wearing an alpaca sweater even, right now, that I got. And I showed yesterday, I showed everybody my Uruguayan hat. But I haven't showed you yet. This is, now, that's the ensemble.

Steve: [Laughing]

Leo: Anyway, security updates.

Steve: Okay. So the only real big news is that anyone who is still using or needs to use Java on their system needs to update it. It was just moved by Oracle/Sun, a major update from them, to Java 6 Update 24. It fixed a large collection of vulnerabilities, in total 21, 19 of which can be used to remotely install malicious software. So it's important. And I did get a kick out of seeing now sort of the wisdom out there, I was reading other people saying, you know, since Java seems to be having so many problems now, and it's surpassed Adobe in vulnerabilities and exploits, removing it, unless it's needed, would probably be a good idea. And I'm thinking, hmm, where have we heard that before?

So, of course, that's something that I have said a number of times, is that sometimes Java gets installed, you remove what it was that brought it in, and it stays behind. The fact that it's in your system allows your browser to engage it. And that's what the bad guys are using in order to now perpetrate browser-based exploits against your system based on Java, which very much like something else like RealMedia, for example, that you'd like to not need, or not have installed unless you're really using it, can be used.

Leo: There are some people, a lot of people installing Java these days for something called Minecraft, which is a really popular game with tens of millions of players. So there are some reasons to install Java. But I'm surprised because wasn't that the promise of Java, that it would be sandboxed?

Steve: Well, yeah. And if the code was perfect, then that's what you'd get. And of course that's why, when Adobe announced sandboxing in Adobe Reader X, it was like, okay, good. But it's like, that's not going to solve all the problems. I mean, the problem is it's little mistakes that are made. And one thing is really interesting about these particular exploits, and that is that in some cases the exploits are machine independent, meaning that traditionally, when we've had a buffer overflow, it's been Intel code, which the processor would execute in the buffer. All of the things we're talking about traditionally have been extremely machine dependent.

But what Java brings you is a machine-independent interpreter that, I mean, that's one of the whole points of it, is it's an interpreter that is able to execute the same Java code in a platform-agnostic fashion. Well, what's a little funny is that some of these exploits are machine-independent exploits, meaning that they could do their bad stuff even on non-Intel-based machines. So it's like, yikes. So, but yes, clearly, if you need Java, you need it. But there's certainly a window, a subset of people who have it just sort of around, like oh, well, you know, maybe I'll need that some day.

And the fact is, if you've got it, you really need to keep it updated, as is the same case with all of our browsers and add-ons and so forth for browsers, because that's the new vector for exploitation these days. So anyway, it's update 24. Probably your Java installation will check for you. You'll get a little square icon down in your tray telling you that it wants to update. But if it's something you know you have, you might want to make sure that you've got Update 24 running.

And hot off the press this morning was news of a new BIND vulnerability, BIND being the preeminent DNS server. Now, this is not a vulnerability like we've talked about before where web spoofing can occur. But it is a problem for existing versions of BIND. It affects versions 9.7.1 through 9.7.2-P3. And there is an updated version 9.7.3.

What happens is, in any high-performance server, and a DNS server is a server just like a web server, for example, where requests are coming in on a continuing basis at various rates, but more or less like the servers being flooded with requests, which is the case with typical fast DNS servers. You have a queue of requests which are in a buffer. And that queue is filling and emptying, depending upon the instantaneous number of requests coming in. Then serving that queue in a state-of-the-art execution model, you'll have what's known as worker threads. You'll have multiple workers which are conceptually each coming back to the queue, getting the next item that needs to be served or serviced and then going about doing their job. And they're all sort of peers. They're peers of each other.

Well, one of the things that can happen, unless the design of the system is exactly right, is known as a deadlock. A deadlock can occur when there are resources in the server which only one thing at a time, by virtue of the nature of the resource, only one thing at a time can access. For example, say that you had something that just wanted to increment a variable. If it reads the contents of memory into a register, and then increments the register and writes it back, it will increment that content in memory.

But imagine that in a really busy system you had multiple threads, multiple paths of execution; and two happened to be trying to increment that at the same time. The first one reads the value from memory into a register. And at that instant what's called a "context switch" occurs. That is, that thread has used up its time. And as we talked about before, the processors aren't constantly doing things or aren't actually doing multiple things at once. They're just jumping around, switching between multiple tasks or multiple threads very quickly. So conceptually we see them as all happening in parallel. In fact, they're time sharing. They're swapping between them.

So this first thread has read the contents of memory into a register and just run out of time. At that instant it's suspended, and another thread is then allowed to run. Well, say that it wants to increment that value. It reads that value out of memory and increments it and writes it back and goes about its business. Well, when the thread that was first doing it is reawakened, it continues. It's got the value that it read already, except now that value is obsolete because another thread, while it was sleeping, came along and incremented it. But it doesn't know that. It's got the value that was current when it was suspended. So it increments that and puts it back, essentially overwriting the value that the other thread wrote.

So what happens is, in modern operating systems, there's a way to handle that. You're able to - a thread is able to declare that it needs exclusive access to something. And while it has exclusive access to an object, it could be memory, it could be a structure, it could be anything, nothing else is able to acquire exclusive access. So that two threads might both say I need exclusive access to this region of memory. And only one of them would have that granted because they both can't have it at the same time. Which suspends that other thread while it's waiting to get its exclusive access. So the first thread does its work, returns the value, and then releases its exclusive access that allows somebody else that might have been waiting for it to run.

But you have to be very careful in the design of these systems because imagine that there were multiple things that a process or a thread needed to have all at once. And imagine that one thread might get a couple of them and then need to wait while it gets a couple others; whereas a different thread might have gotten some of those other ones, and it's waiting till it gets a different set. The point is it's possible for two threads to each have something that another threads needs, and neither of them to be able to move forward until they get them all. And that's a classic deadlock in computer science. And...

Leo: Is it what they call a race condition?

Steve: A race condition is sort of what you would have if you didn't have this protection, where you still get, like, an inter-thread competition, but it's sort of in a different fashion. So what has happened is that the guys at Neustar found a deadlock condition in BIND such that there's a process called an "incremental transfer," where one DNS server is able to ask another DNS server for a bunch of data about a zone that it's managing, a zone being the technical term for, like, a domain, essentially.

And it turns out that there is a short window of time where, if you asked for an incremental transfer, and made a query of the DNS server virtually simultaneously, you can get that BIND server, the entire DNS server to lock. That is, it's exactly this problem. It causes a deadlock. There's just a tiny little mistake that they made somewhere in their code such that, if those two types of requests come in virtually, are being handled virtually at the same time, the entire server locks. It just, if you have - it stops servicing any DNS requests.

And so it is a classic golden goose for the bad guys, I mean, because BIND is the server everyone is using. This is the latest version. The latest version, I mean, like, except for this very, very latest one, because this announcement just came out this morning. So the previous release, that was current, can be locked up solid. I mean, you've got to stop the server and restart the server in order to get DNS services going again. And so it's regarded as a high-severity advisory for BIND.

And one thing, there is a fun little workaround. If you can't upgrade immediately to 9.7.3, but you can restart your server, you can restart it with a "-n 1" option on the command line. And that says run a single worker thread. So it literally, the -n command tells the server how many worker threads to run. And if you tell it only run one, then there can't be another one, there can't be two threads that are trying to access the same resource, so you can't have a resource contention problem, and it won't lock up. But it's only if you have more than one thread that, in this version, in this prior to 9.7.3, you can get a condition where there's a deadlock, and two threads are both trying to get something while they own some resources the other one needs, and nobody can move forward. So, interesting little bit of computer science that has sort of come back to bite the developers.

Leo: Yeah, I love that. Well, but, now, this is temporary. You're not going to want to do "-n 1" forever.

Steve: No, and, see, that's the problem is this notion of a worker thread pool is state-of-the-art maximum performance because what can happen is a thread can be doing some work, or making a request of a back-end database, or there are many things that can happen that causes a thread to stall, like it needs to wait for something. So you'd like to have, in a busy server, many other things that could be done. While one thread is waiting for some piece of work to complete, other threads get control, and they're able to move their own little bits of work forward.

So literally it's like having a team of workers who each go back to a queue, get their next job, and then wander off and start pursuing it. And someone might take a lunch break, well, other workers are still going. So it's a nice asynchronous model which is very effective for getting maximum work out of a server. So a busy DNS server probably relies on having a pool of active workers. And if you just said, sorry, everybody's fired except Joe, then Joe might not be able to do nearly as much work as the whole group.

Leo: I had no idea that my programs were having lunch breaks. But that's good to know. Get back to work.

Steve: So we did have some congressional testimony this last week from our friend Valerie Caproni at the FBI. This is on this whole going - what they call their "going dark" problem, which is what the FBI has named their increasing concern that their practical ability to wiretap the Internet is slipping from them because more and more of what's on the 'Net is encrypted. And of course we've been promoting it, for example, in the wake of the release of Firesheep. We've been saying, wow, you really want to be using SSL and HTTPS communications so that the bad guys can't be sniffing your traffic in wireless hotspots. And we celebrated when Facebook recently added that option to their configuration, finally allowing, as they had said they were going to, a user to say "Force HTTPS secure connections wherever possible," which is a great move forward.

Well, what's good for us and our privacy, of course, is bad for the FBI and our other security and intelligence agencies that really feel that they need the ability to be able to see into the traffic on the Internet in order to protect us. So the problem is that the FBI is still being very circumspect and cagey. I mean, I was impressed with the testimony. I was impressed that the people who were in this panel seemed to have enough of a grip on what's going on to make this useful.

CNET - actually this was widely covered, The New York Times, The Wall Street Journal, CNET and others because people are wondering what's going to happen with this legislation. Of course I have been wondering because I'd like to do a VPN technology which is explicitly for the sake of protecting us from bad guys. And I'm uncomfortable with the idea that there might be some legislation coming downstream here that says, no, sorry, we need backdoors in anything that uses crypto.

And of course the problem is that the FBI has still not articulated what it is that they want. They've mentioned Skype, which concerns me because Skype is point-to-point encryption. As we know, right now, Leo, you and I have a direct connection between us, between my machine and your machine. It was mediated by Skype. Skype did the presence management to show us each other and allow us to find each other. But our connection is point to point and powerfully encrypted. Skype has a very good crypto technology. And so there isn't any way for our dialogue to be eavesdropped on, whether it's text chatting, or audio...

Leo: Not that anybody would really want to, since we broadcast it live. But all right.

Steve: But of course the bad guys are saying, hey, now we know how we can talk to each other without worrying about being overheard. So in CNET's article they said, "FBI general counsel Valerie Caproni will outline what the bureau is calling the 'Going Dark' problem, meaning that police can be thwarted when conducting court-authorized eavesdropping because Internet companies aren't required to build backdoors in advance, or because technology doesn't permit it. Any solution, according to a copy of Caproni's prepared comments obtained by CNET, should include a way for police armed with wiretap orders to conduct surveillance of web-based email, social networking sites, and peer-to-peer communications technology."

So I've listened to the testimony. I've listened to the news reports afterwards. And it's still not clear what it is they want. We really didn't get much from it. The EFF weighed in. They've got some documents that were just released that they got under the Freedom of Information Act, which is still sort of murky. What I'm hoping is that - and from some of the things that have come out, it sort of sounds like this is reasonable - that the problem the FBI is addressing, for example, is that they'd like to be able to go to Facebook and Google, with whom they have held talks already, and be able to serve them with a court order, a wiretap surveillance order, and then be able to receive a stream of some sort from the service providers on the use of certain individuals of their services.

And so what Facebook and Google are saying at the moment is we don't have that built in to our system. Yes, we could do it because we're the database. We're one end of this connection. And so, yeah, that information is here. But we don't have the technology to, like, tap ourselves. We haven't ever needed to, and frankly we haven't wanted to. And so what it sounds like, again, sort of reading between the lines, there have been comments made that, like, well, we realize that when individuals encrypt their own communications for their own sake, that's something we can't get to. But when services like Facebook and Google are doing so, well, we need that, and it's reasonable for us to have it. So it sounds like what they're trying to get is some legislation which would require anyone who is like a receiver, I don't even know how they would describe it legislatively, but a public entity like...

Leo: A carrier.

Steve: A carrier, well, but sort of an endpoint. See, the problem is...

Leo: No, that's true, yes, they're not just a carrier, are they.

Steve: Yeah. So an ISP can't decrypt VPN traffic. They just don't have the key. And but someone at either end does.

Leo: Right, the endpoint does, yeah.

Steve: The endpoint does. And so there was a comment made that in some cases our law enforcement would simply have to find other means, meaning that they are recognizing there are some things they just can't get. And from conversations with Google and Facebook and, I mean, just we understand the way the technology works, there are things they could get, except that there isn't the facility for it now. And so it sounds like what the FBI would like to have is some legislation to force entities like Facebook and Google to be able to respond to a court-ordered wiretap when it's provided. At the moment they can say, sorry, we'd like to help you, but our system doesn't do that. And so there would be legislation that would force their systems to have that added to it. And I think that's where we're going to end up. Which is - I hope it doesn't get abused, I guess, is all I'm saying.

Leo: I was reading an article, I guess we talked about this before I left, about these pen register warrants, which do not need to be disclosed, and they're not as strongly regulated because it's presumed, well, there's no content being revealed, it's just the fact of communication, the subject of the email, your GPS location. And these are being heavily used. I got some emails after talking about it on the radio show from law enforcement people, said, oh, yeah, we use those all the time.

In the U.S., phone companies are allowed to profit from these requests. They charge the police a few bucks. In fact, Sprint has a web portal for law enforcement. You want to know where somebody is? No problem, you don't need a warrant. Just say you're law enforcement, give us the person's phone number, and we'll tell you right now. In fact, we'll tell you forever where they are. It just costs you a couple of bucks. In Canada it's illegal to do that. So they have to give it to you free. They still do it, but they just can't charge for it. So it really does feel like our privacy is being eroded very rapidly.

Steve: Well, speaking of which, we have a couple more little bits that I wanted to share with our listeners. And you're exactly right. The COICA is the acronym for Combating Online Infringements and Counterfeits Act.

Leo: Oh, I hate this.

Steve: I know, and it's back.

Leo: It's back. Great.

Steve: Or it will be shortly. It was introduced last year, but the Senate did not take it up. And it's being reintroduced with somewhat better controls to limit what the DoJ can do with it. And one significant restriction which is being added to the legislation which is trying to be reborn here would be that domain seizures could only be used when less restrictive methods have failed. And so basically, this is what we've talked about a couple times, Leo, where...

Leo: That preemptive thing.

Steve: Yes, well, and where basically the MPAA, the RIAA, the large, powerful, lobbying content owners are saying, we need ways of getting counterfeiting websites shut down. We just want them gone.

Leo: And we don't need that due process thing. That's so old-fashioned.

Steve: Well, and due process, exactly, you're right, Leo, due process is the problem. Senator Sheldon Whitehouse, who's a Democrat in Rhode Island, he was quoted saying, "I contend that America is on the losing end of the largest transfer of wealth through theft and piracy in the history of mankind. We're doing virtually nothing about it."

Leo: I love it how these guys are so hyperbolic. Oh, my god, the record and movie and TV industry, how will they survive?

Steve: Yeah, yeah. Since late November, the U.S. Department of Homeland Security's division, it's called ICE, the Immigration and Customs Enforcement, that's the group that does this, they've obtained court orders to shut down more than 100 websites for alleged copyright infringement, even without this new authority which the COICA would give them. And then just Monday of this week they announced they had seized the domain names of 18 websites which were offering counterfeit jewelry, handbags, perfume, and other products. And this is something, though, that seems to have some global sweep to it, also, because Spain also just passed a similar law. They had tried to pass it also last year and failed. It was voted down. They then added a panel to oversee the shutdowns, basically an oversight committee, and with that addition Spain was able to get the law passed.

So on that note, another bit of news is that one of these ICE, Immigration and Customs Enforcement takedowns completely backfired. They were attempting, this was also last week, they were attempting to seize 10 web domains suspected of storing, displaying, or peddling child pornography. Unfortunately, in the process, they also seized a site called Mooo.com, which is the most popular domain at Afraid.org, which is run by the DNS provider FreeDNS. There are 84,000 subdomain websites hung off of Mooo.com, and they were all taken down.

Leo: And so presumably this is some sort of hosting company or web hosting.

Steve: Exactly. Exactly. It's a huge web hosting company. More than 84,000 companies and individuals had websites hosted there. And they were not only taken offline, but instead, visitors to any of those 84,000+ websites were taken to a page with a banner that said this domain - it shows the logos of the Department of Homeland Security and the Department of Justice. The banner says, "This domain name has been seized by ICE - Homeland Security Investigations pursuant to a seizure warrant ... under the authority of Title 17 USC 2254. Advertisement, distribution, transportation, receipt, and possession of child pornography constitute federal crimes...."

Leo: Talk about killing a mosquito with a cannon.

Steve: So all of these sites were accused of trafficking in child pornography. And it took 48 hours for this mistake to be corrected at the DNS level because essentially the registrar for Mooo.com was forced to change the root level registration to point to a server that presented this notice. So all the subdomains that were hung off of Mooo.com received this notice. And due to Internet DNS caching, it took a total of about three more days after that for this all to get itself sorted out.

And one blogger who had this happen to his site, who was obviously completely innocent of any of this - ICE is being overseen by a guy named John Morton. And so he blogged: "Mr. Morton, with all due respect," and then we'll blank out this expletive. But it was a word...

Leo: Eff you.

Steve: Uh-huh.

Leo: In the words of Cee Lo.

Steve: Something off. He said: "Get out of my Internet. You'd get no argument from me that there are truly distasteful and illegal things on the Internet. That's true of any society. But there are also proper ways to deal with these problems. Pulling a total domain, sweeping up innocent people along the way, feeling that you don't have to comply with due process of law, and indicating that you don't give a damn is wrong. It's not as wrong as child pornography or counterfeiting, but it's still wrong. As a taxpayer, I feel you're wasting my money and denying my ability to use the Internet to host a server containing useful, legal, and hopefully interesting content over a readily known alias.... That's to say nothing of any damage done to my name or reputation by this idiotic law." So, whoops.

Leo: Well, yeah, no kidding, whoops.

Steve: Yeah.

Leo: Yeah.

Steve: So apparently what happened was there was one of those 84,000 subdomains of Mooo.com was the actual bad guy target. But again, here's the problem, is Internet technology is complicated. And it's important for the people who are going to be given this kind of power to be technically competent, to understand that they want to remove a subdomain where the subdomains are all hosted off of a primary root domain. So we understand this would have been, who knows what, fancypursesforyou.mooo.com. That might have been a piracy site. But that's a subdomain off of Mooo.com. Unfortunately, the DoJ killed the root domain that 83,999 other good sites were pointed to, and aimed them at this disturbing page so that anyone going to any of those sites would have seen something accusing them of child pornography. So, yeah. Clearly this was a mistake. I read some nonsense on the 'Net about this being deliberate. It's like, well, there's no way this was deliberate. That's crazy. But it does say that those who are doing this...

Leo: Don't know what the hell they're doing.

Steve: Yes.

Leo: That's what it says.

Steve: They're going around stomping, at the registry level, stomping on root domains.

But in the case that, exactly like this, this is a hosting site where you've got a huge number of subdomains, each containing separate websites, you just can't go kill - it'd be like them killing off .com, you know, what would happen if they killed off .com? Well, it's clear that we would all cease to exist.

Leo: Not me. Not me, I'm TWiT.tv.

Steve: You're .tv. Leo would be the beacon.

Leo: I'd be the last guy standing. I have Leoville.com is the only .com I have. Almost everything else is a dot something else.

Steve: I'd just be on the roof waving.

Leo: Hey. Hey, over here. They cut off the Internet.

Steve: And then in the last little bit of sad news - I don't know if this is sad news. I don't know how anchored you were to U.S. politics during your cruise, Leo.

Leo: Actually, I missed - I was watching Egypt, but I missed what happened here. What happened?

Steve: What happened here was that, early, early, at 4:30 a.m. this last Sunday morning, the House of Representatives passed their H.R. 1 bill, which is very controversial. This is where they are taking $61 billion off of our current budget in order to get us to September. And we have until March 4, which, unless this ends up getting through the Senate and not vetoed by our President, then the federal government shuts down I guess at the end of the day of business on Friday, March 4th. One of the things that they stripped out...

Leo: Snuck into it.

Steve: They stripped out a lot. I mean, Planned Parenthood lost its funding, and the EPA lost its ability to enforce greenhouse gas emission controls and all kinds of things have been removed, regardless of how you feel about it politically. One thing also that got removed was that the FCC lost their ability to spend any money on implementing their Net Neutrality rules which they had proposed last year. Now, it's still not clear whether the FCC has the legal authority, and Verizon has challenged them, is challenging them in court, whether they have the legal authority to enforce Net Neutrality. But in any event, they don't have the budget any longer, or they may not if this stays in what finally gets passed.

And there are lots of vulnerable programs that I think, even if there's some compromise that's reached, that backs off of the $61 billion that's being cut. Things like this, I imagine, I won't miss it. We need some, as we've talked often, we need some sort of legal precedent for how Internet carriers can regulate traffic and what the limits of them doing so are. So we're just sort of stumbling forward, trying to figure out what we should do.

Leo: You know, it's really, I mean, if you know how to manipulate the government and the legislative process, there's not much you can't do. I mean, just tie it to a bill that will shut the country down if it's not signed into law, and what can anyone do?

Steve: Yeah.

Leo: Amazing.

Steve: Yeah. Now, you did miss a fantastic episode, unfortunately...

Leo: Oh, shoot.

Steve: ...two weeks ago. We discussed something called Bitcoin.

Leo: Oh, I'm familiar with Bitcoin.

Steve: Oh, good.

Leo: I actually run a little Bitcoin server. Nobody ever gives me any Bitcoin, but at least I earn it myself with my little server.

Steve: We completely covered all of the technology of Bitcoin.

Leo: Isn't it interesting? I mean, I don't know if it's going to take off, but I think it's very interesting.

Steve: It's fantastic. I'm a fan. And we really had fun a week before last talking about it, and then last week was the Q&A that had about half of its questions follow-ups. And what I wanted to tell our listeners was that - I made the comment last week that, when I'd had my little Bitcoin server running, that of course nothing had happened. Well, I turned the screen on, and I found out I had won 50 Bitcoins.

Leo: Right. They do that to keep you going every once in a while.

Steve: Well, no. Well, actually the chances are about that everybody doing it would, as a function of the amount of processing power in the network, it would take about a year. And it was on Valentine's Day at 7:32 p.m. that my computer - in fact, Leo, it's that i7875. And it's only because I built that little powerhouse in order for us to play...

Leo: You've got a lot of threads there, baby.

Steve: ...that I got, exactly, I'm able to do 4,800 hashes per second. And so just - it's just pure luck, basically. But I was out there pitching my hashes out onto the peer-to-peer network, along with all the other Bitcoin servers on the peer-to-peer network that were trying to solve this puzzle, and I got one first.

Leo: And we should point out that this is completely automated, that it's not somebody rewarding you for covering Bitcoin. Nobody pushed a button at Bitcoin headquarters or anything.

Steve: Oh, and no one's...

Leo: There is no Bitcoin headquarters.

Steve: Well, yeah, and it's anonymous, too. There's no way that...

Leo: Right, they wouldn't even know.

Steve: It's not like they thanked me for covering the podcast, I mean, on the podcast for covering the technology. They have no idea it was me. It was just pure blind luck that on Valentine's Day my machine happened to be first in solving this hash puzzle, and I got 50 Bitcoins.

Leo: The real issue is what the hell can you do with them.

Steve: Well, yes. It's just fun. I mean...

Leo: You can give them to me. I'll take them.

Steve: There are now - the current trading rate is about one bitcoin per dollar. So I can cash those in.

Leo: There are people who are actually - there is a market.

Steve: Yes, yes. You can buy services and goods. The EFF accepts donations in bitcoins.

Leo: In Bitcoin? Oh, that's great.

Steve: In Bitcoin.

Leo: Because I've been giving them American dollars. I'd better start giving them Bitcoin instead.

Steve: Oh, Leo, that's so old-fashioned.

Leo: Old-fashioned.

Steve: Dollars, no. We want virtual money. We want crypto currency.

Leo: I just love the idea. It's just a great idea.

Steve: It's pure anarchy. And in fact, some of the questions that we talked about last week were people were worried about, well, can't you use this for money laundering? It's like, absolutely. And you can also use it for privacy, if you would like it. It's double-edged, unfortunately. All technologies are.

Leo: How would you money launder with it?

Steve: Well, you could easily take currency in any denomination and convert it to bitcoins, send it to somebody else...

Leo: Oh, that's why there's a market.

Steve: ...and have them convert it back out.

Leo: I couldn't figure out why somebody would give you actual money for bitcoins, and now I understand why. I set it up as an experiment. And I thought about using it for - somebody wanted to donate bitcoins, so I set up the server and all that. And maybe we'll take bitcoin donations. I just - I don't know what you do with it. It's just kind of interesting.

Steve: It's just sort of a trophy. I got 50 from having my machine on the 'Net. That's kind of cool. And people have asked, would you be willing to sell your software, Steve, in return for bitcoins? And I said, well, I'm not set up to do that, but next time I go in to my eCommerce system, I'll think about maybe...

Leo: Maybe eat bitcoins, I might. Pay the rent with bitcoins, I might. I have to say I'm glad to hear, though, that you vetted it and went through the process that they're doing, and it's...

Steve: Oh, Leo, it's so cool.

Leo: Yeah. I'll have to listen to that episode.

Steve: Oh, they just nailed it. I am so impressed with the way the system works, the fact that everything anyone's been able to come up with was incorporated into the solution. I mean, it's one of those things where it really works. And it's self-sustaining. And it's taking off. And the value of bitcoins has been going up. And in fact we apparently created a small denial of service, that is, the podcast discussing Bitcoin.org took it off the 'Net for a while. There was a notice up saying that, due to overuse of the server, it was like a redirection somewhere else. So we may have just used up their bandwidth.

Also, last week or the week before, we had talked, I think, about how Symantec had purchased VeriSign's ID system, the VIP. And I discussed, I think it was also last week, that the news that the forthcoming chipset from Intel, the Sandy Bridge chipset, would incorporate VIP technology in it, which is very cool. So that the little football that we've talked about and the eInk credit card and all that, that used the six-digit code, Intel has built that technology into the hardware. So basically your laptop or your desktop, whatever it is that uses one of the Sandy Bridge chipsets that contains this, will itself be an authentication token. It has the crypto stuff in it. And once you register that, in this case with VeriSign or Vasco or one of the providers, then you'll be able to use it for authentication, just like you do the football or the credit card.

The reason I bring this up again is that I was assuming that - I was, like, wondering, well, okay, we were also discussing Google's move to multifactor authentication. And I was lamenting the fact that Google had not, among their many things they do support, like a cell phone can call you and read you a code that you then type in, or you can get it via text and so forth, Google has basically unveiled multifactor authentication for their stuff now. I was wishing that they - I said, yes, but they don't support, darn it, VeriSign's VIP system. And then I thought, well, maybe it's because it's not free. Maybe people who use it, we end users don't, but maybe the sites that are using it for authentication do have to pay for it. Like, for example, PayPal and eBay are paying Symantec something for the use. And I did get email from someone at Symantec who kept himself anonymous but said, Steve, you were right. It is a service which is paid for by the websites that use it for their authentication.

Leo: Oh, interesting.

Steve: So, yeah.

Leo: That explains that.

Steve: Explains it, and also makes the economic model make sense, too, because, as I had mentioned before, I now have the VeriSign Symantec VIP applet on my BlackBerry. I know that it's available both for Android and for the iPhone, as well. So it's no longer necessary for you to have a physical football, or even a credit card. If you've got your phone with you...

Leo: I know, I love that, I love that.

Steve: Yes, you can now authenticate in there.

Leo: That's so much better, really. Because everybody always has their phone.

Steve: Exactly. It's the perfect solution. And I do have a neat note from a listener of ours, Philip Garrett, who wrote, in my quest to always find new SpinRite stories, "SpinRite Saves a PS3." Of course the PS3's been in the news a lot lately because of the backdoor stuff that's been happening, all of the hacking of the PS3. He said, "Sir or Madam: My PS3 Slim started acting flaky during a game add-on download on Sunday, February 20, 2011. The interface was sluggish and would momentarily freeze up. On Monday evening, when I arrived home from work, I attempted to turn the PS3 on, and it would only perform an incomplete boot before hanging up and freezing solid.

"I have two years' worth of game-saved files and other moderately important information on that hard drive, so it was important to me to repair it. I had a hunch that the problem was with the PS3's hard drive. I removed the hard drive from the unit and slaved it to my main machine. I inserted my SpinRite disk and rebooted the machine. SpinRite booted right up, and I was able to select the slaved drive and run SpinRite at Level 2. After 30 minutes, SpinRite completed its work and showed one unrecoverable sector. I crossed my fingers, hoping the drive had been repaired enough to properly boot. I placed the drive back into the PS3. The PS3 boots right up. I was able to immediately perform a backup of the data onto an external drive."

Leo: Good man.

Steve: "No longer having any faith in the original drive, I jumped into my truck and left for Fry's. I was able to purchase and install a new drive. After formatting and reinstalling the OS, I was able to restore my backed up data. My PS3 is back to running like a dream. Thank you for reminding us on Security Now! that SpinRite is not just for computer hard drives. I can now say from experience that it works on a PS3 drive. Thank you again for a great product. Philip Garrett in Fishers, Indiana." And thank you, Philip, for sharing that.

Leo: That's great news. All right. Time to talk proxies, Steve Gibson.

Steve: So normally when we use our web browser, we put the URL we want of a site we want to visit into the address bar. And as we know, the browser looks up the IP address of that website, that domain, and then attempts to initiate a TCP connection to that IP on, by default, port 80 if we're just using HTTPS, or port 443 if we're using - if we're just using HTTP, not HTTPS, or 443 if we are using HTTPS. So the browser connects directly to there, to its remote IP, and that standard web surfing port 80. And then, if it's able to get a connection, exchanges its request with the remote server and obtains whatever resource, web page or whatever, it's asking for.

Now, there are some cases where it's useful to add a layer of complexity to that. A famous instance which most users are not aware of is when an ISP is proxying connections on behalf of their own customers. In that case, the web browser thinks that it is connecting to a remote server, but in fact that connection is intercepted by the ISP's caching proxy, which looks at the request to see whether it might have what the browser is asking for in its own cache.

In the case of very popular sites like Amazon, for example, that are covered with menuing, images, and just all kinds of stuff all over the page, it's very likely that that same stuff is being delivered to all of the customers of the same ISP. And so if this intercepting proxy saves those when it retrieves them for one customer, it can save having to retrieve them for someone else. So that saves the ISP bandwidth going out to the Internet, and it arguably improves the customer's experience because they're going to get their own page loaded much faster because their browser is not actually having to go out onto the Internet to get, like, all of the extra stuff on a page.

So this notion of a local caching proxy, which is transparent, is one which users don't normally see. There are other instances, though, where the proxy is either automatically configured or manually configured to serve a nontransparent, some specific purpose. For example, you might have a corporation which wants to control the exterior or external use of the Internet by its employees. So in such an organization, the organization's firewall would block outgoing connections to port 80 and port 443, so that all of the web services that exist outside of that corporate network are inaccessible. Those servers are serving their content on port 80. But if your web browser is unable to send Internet traffic out destined towards port 80, it can't get on those websites. Another instance might be schools or universities or corporations that don't want to do a wholesale blanketing, but for example they want to keep their employees from spending all day on Facebook or logged into Twitter. So there they're blocking specific websites at specific locations.

And of course the much larger instance is a country where there's a government like China that sort of officially intends to censor those websites that its citizens are able to access, in which case they've got industry-strength firewall technology running at their national borders which are preventing anyone from, when Google turns up search results for links they click, if those are on sites which are proscribed, they're not able to connect to those servers.

So a proxy provides a means for providing sort of a middleman, which is exactly what it is, in the connection between a user's browser and the server they're trying to access. In the corporate access case, you might have to log onto the proxy server in order to gain access outwards. So you have to authenticate yourself, essentially declare your interest in going to the outside world. And that gives the corporation a means for controlling your access. It might be monitored; it might be logged. They still might be filtering where you can go, even when you log into the proxy server in order to get out.

But the concept of proxying, from a technology standpoint, is one where, if the proxy is configured in your browser, that is, your browser has been explicitly told that it will not be able to get directly out onto the Internet, it has to use a proxy, then no matter what address you put into the address bar, the browser connects somewhere else. It doesn't look up the IP address of the domain you put in the address bar and attempt a connection. Instead, those settings which are in its configuration dialogs, those take precedent. And so, for example, there will be an IP address or maybe a domain name and even a port number that essentially completely overrides the connection level part of this dialogue.

So the browser always goes to a specific IP and port number. That's what it connects to. And if it is authenticated, if authentication is required, then what happens is it makes its query, just as it normally would, as if it had normally connected to the actual destination IP. However, it's connected instead to this proxy server. And the reason the name "proxy" is that this server then acts on behalf of the web browser outwards toward the Internet, that is, it proxies its request and makes it on behalf of the web browser. So what this allows is, it allows control to be applied, both outgoing and incoming.

Another example, when I was poking around looking for some good examples, I ran across UCSD, UC San Diego. They have instructions on their website for students of the university who are outside of their university network. And they talk about being on AOL or being on Cox or being on some other carrier, but who want access to, for example, the university library system, which is on their internal network. UCSD has many servers running inside its network which are not available to the general public out on the Internet externally. So if their students are working, for example, off campus, on Cox, that's not within the university network, yet the students may need access to those resources.

So students can configure their browser, putting in webproxy.ucsd.edu as the domain where the proxy is; and then the web proxy port in the case of UCSD, and this is just arbitrary, is 3128. So a web browser running outside of UCSD, when attempting to connect to resources inside UCSD, is able to essentially redirect all of its traffic to the IP of that domain name, webproxy.ucsd.edu, and not connect to port 80, connect instead to 3128, and then they have to authenticate. They've got to be a student with UCSD login credentials. When they try to make any connection there, sort of like when you use a hotspot that isn't completely open, where you've got to jump through some hoops first in order to log into it in order to get access, the same thing happens here where, instead of trying to get out, you're essentially trying to get in to an internal network that's protected that way. So the idea is that, if you're authenticated, then the server that is answering and fielding those requests from the outside is able to turn around and send that request into the interior network.

Now, the problem with this sort of a static proxying is that your browser sends all traffic there, not just traffic bound for servers and services that are inside UCSD. So there's another fancier way this can be done, and that's with a proxying script. If you look in your web browser, whether it's IE or Firefox or any of the others, there are normally a number of options for configuring your browser proxy. And in fact one of my favorite tips for people using IE, whenever I go to someone's house and they've got some problem and say, hey, can you take a look at my computer, if I fire up Internet Explorer, and it takes a long time, I go, "ough," and immediately go into their Internet options and turn off, under LAN settings, automatically detect settings. This is a really annoying thing that Microsoft has always done with Internet Explorer that delays its startup every time you launch it.

Leo: I go "ough," too.

Steve: Ough. So the Security Now! get-on-the-'Net-faster tip of the week, if you're using Internet Explorer, is under Internet options, connections, LAN settings, you'll find that, unless you turn it off, it'll say "automatically detect settings." Now, Firefox has the same default, but for whatever reason I've never felt the same delay. When I went to look at my Firefox configuration, I found it, too, had it turned on, and I immediately turned it off. Maybe Firefox is a little less slow in handling it, or maybe it does a better job or does these things in parallel. Because what this automatic detection does, it's a sort of a kludge protocol which allows your browser, knowing nothing at all about a network, to determine whether it needs to use a proxy in order to get out on to the Internet. So this was one of those things where Microsoft decided, okay, even though this is probably only useful for 0.2 percent of the users of Internet Explorer, we don't want to get phone calls or customer service problems from those 0.2.

Leo: We don't want to turn it off because, yeah.

Steve: Right. So we're turning it on for everyone, even though now 98.8 percent of the Internet users in the galaxy are all going to go, "ough."

Leo: You're right, because actually that's what I do, too. Because you know it's checking to see if there's a proxy.

Steve: Oh, and it's a slow process.

Leo: It is.

Steve: The first thing it does is it does a DHCP broadcast. We talked about DHCP, Dynamic Host Configuration Protocol, which is the way our computers find their IP address and subnet and so forth. Our routers are DHCP servers. So it's a neat technology for allowing systems to configure themselves. And I have mentioned also that DHCP can supply many other kinds of information. You could get time of day from it, if it was configured to offer it, and all kinds of other information.

Well, one of the kinds of information you can also get from DHCP, in addition to give me an IP address and my gateway IP and my subnet mask, is, is there a proxy here, a web proxy that I should use? So you make a DHCP request of option type 252. And if that is supported on your DHCP server, it'll respond. Now, when the browser makes the request, it sends it off and waits for a reply. And if it doesn't get one, it sends it again a couple times because it might have lost the first response or the reply. If that doesn't work, then it falls back to something called SLP, which is Service Location Protocol, that almost no one has, but somewhere someone had it once. So Microsoft says, well, maybe they still do, and so we wait for that, too.

Leo: This is the curse of Microsoft is this backward legacy for anything anybody ever did.

Steve: Yes. And then, even though almost no one has this, they say, well, let's check DNS. So have you noticed, Leo, and I've always wondered why my Windows wants to know what, like, the computer's own domain name is.

Leo: Right.

Steve: And it's like, my computer doesn't have a domain name.

Leo: It may not just be DNS. It might be there's WINS and there's other systems incorporation. It's a business thing; right?

Steve: True. So what happens is, if your computer has a domain name associated with it, and there are corporations where, for example, when you get online, your machine will have a name, and so it'll be your machine name dot Jimmy's Hotcakes Corporation dot com.

Leo: They definitely use Windows.

Steve: I'm sure they do. And so what happens is, if nothing else has responded yet, then IE puts the prefix "wpad" - which stands for Web Proxy Auto Discovery - it puts that in front of your machine name. So it would be wpad.mymachine.jimmyshotcakes, or whatever I said, dot com. And it does an address record lookup for that. And of course, if it doesn't respond, it tries it a few more times until it's sure that it's not there. Then it does an SRV record lookup and waits for that to give up. Then it tries a TXT record lookup and waits for that to give up. And if that doesn't work, then it shortens the path by removing your machine name and just tries wpad.jimmyshotcakes.com and does all of that again. So this is why people are going "ough" all over the place, is waiting for IE to get going, this is what's happening. So by all means, if you've ever noticed this, or even if not, and you're using Internet Explorer, which seems to me the slowest of doing this for some reason, Internet options, connections, LAN settings, automatically detect settings, turn it off. I mean, unless you need that. Some people probably do.

Leo: Oh, if you're in a corporation you probably should not do this.

Steve: Well, and in a corporation it won't slow you down because...

Leo: Right, because it's going to find something.

Steve: Exactly, your machine will make a broadcast, say hey, here I am, do I need a proxy? Somebody somewhere will say, you sure do.

Leo: Oh, yeah, here it is.

Steve: And here it is.

Leo: Yeah. We used to, I remember ZDTV was set up that way, TechTV. And that made sense in a corporate environment. But at home it makes no sense at all.

Steve: No sense at all, and all of us have it on, and we're all going "ough" and waiting for IE to get going. And a little less so for Firefox. Mine's turned off now for the first time with Firefox. And I can't wait to, like, restart it and see if it's faster.

Leo: I wonder if Chrome is doing - I guess everybody has to do it; right?

Steve: If they don't want to risk compatibility. I sort of thought Firefox might have it off by default. But I guess maybe everybody has to have it on. And so we're all losing some time in this because some people somewhere have to have it on. All the rest of us, unless we've gone in and turned it off, are waiting for that to expire.

So as I was saying, for a user who, for example, needs to get to proxied services - for example we'll take the UCSD student example. If they configure, manually configure their browser to use a web proxy, then everything it does, it sends there, which is a problem. So what that would force the user to do is to be going in and turning this on and off all the time. That is, if they want to go out and surf the 'Net, just go to Facebook and Twitter and Google and everything else, they have to turn off the proxying, which is a few dialogues down to get to, in order for their web browser just to make direct connections out to the sites they want to visit. Then they've got to turn it back on again when they want to go back into the UCSD network.

Not surprisingly, there are some utilities which have been created, one called GProxy, which makes switching the proxying on and off much easier. It gives you a nice user interface for doing this. And there are a bunch of those. But there's one slightly cleverer solution, and that is, if you look at this proxying dialogue in your web browser, you'll see there's the solicitation for a script that you can give it. Now, the bad news is it's got the word "script" in it, and we know how I feel about scripting. It actually is JavaScript. And so in the case, again, of UCSD, there's a file there, webproxy.ucsd.edu/proxy.pl. And I don't know why they use "pl." Unfortunately, that's a common extension for Perl scripts, and this is not a Perl script. It's JavaScript. But if anyone's curious, you can put webproxy.ucsd.edu/proxy.pl into your browser's address bar. It will probably pop up and say, would you like to save the file? You could save it and then look at it.

And what you'll see is a very sophisticated JavaScript program which analyzes - which your web browser can now pick up from, if you are a UCSD student, pick up from UCSD. And it, with very fine-grain detail, tells it which URLs and servers and services and domains and all kinds of stuff, IP addresses, I mean, you have all the power of JavaScript essentially in this filter so that every URL your browser is given is passed through this function. That JavaScript file defines a function called "Find proxy for URL," and has given the arguments of the URL and the host machine name. And it returns essentially a proxy string that tells the browser how to connect.

And so the beauty of using that is that you can still go to Facebook and Twitter and Google and anywhere you may want to because that script will say, nope, we don't handle those domains, go direct. And so your web browser will make a direct IP connection there. And if it is a domain inside UCSD, that script, when it's given that, will say, oh, yeah, here's the settings you want to use. Make your connection to this machine at this port number. And so it makes that process of sort of being in and out of a proxy very nice. And potentially you could alter the script yourself, if there were different proxy servers you wanted to use for different remote sites.

So it's a powerful capability. Unfortunately, it's also JavaScript. And there have been, as one would imagine, exploits where the web proxy auto-configuring script has been hacked by people because think of the power it gives you. Basically, if that were maliciously altered, then your browser is going to blindly follow that script and connect to whatever machine and IP and port this script has told it to, and that's completely transparent to the user. You put in the URL. You don't see where you've really gone. Your browser connected off to Russia, unfortunately, instead of to Palo Alto and Google. But because it got a malicious - it's called a PAC, a Proxy Auto-Configuration JavaScript file. That has happened in the past. So it's just something to be aware of. But still very cool capability, which has always been in our browsers, which most of us are just sort of unaware of.

Now, the final type of proxying - oh, there's two more. The other type of proxying that many security-conscious users have used in the past is a local proxy, where instead of this being a remote server that you connect to, you actually run a server in your computer. Famously, Proxomitron has been used for years. And more recently there's something called Privoxy, which used to be called the Internet Junkbuster, but they ended up folding up shop, and this thing went open source, and it's Privoxy.org is the home of the Privoxy proxy. This is multiplatform, open source. It's something you run in your computer which essentially sets up a server which provides local services. You then configure your browser using the same proxy dialogue which you may have now found, if you've been listening to the podcast so far, because you wanted to not, "ough," wait for IE to start up so slowly every time.

So you configure your browser to use this server running in your own computer. And the reason you do this, the power of it is that it makes a very powerful filter. This is what Proxomitron had been used for for years. Proxomitron is still around, and as far as I know is being supported, and is well used by people who've gotten into the nuts and bolts of taking responsibility for their own security and privacy. One of the things, for example, that people like to do is not declare publicly what their user agent is. The user agent is the header which a browser sends out to talk about what make and model and version. And increasingly, it also has a long string of stuff, of, like, if you've got .NET installed in Windows, which is becoming more and more unavoidable these days, there's all kinds of version information. When we talked about the service that the EFF runs for fingerprinting browsers, Panopticlick, one of the things that Panopticlick uses to lock onto users that make us look so unique is that user agent string because it's got all this version information. It's getting longer and longer. It's just a rich, harvestable source of info.

So imagine that you don't want your browser, your system to be sending out information. Something like a local proxy can fix that because essentially it's in the connection between you and the - between your browser and the Internet, your browser being told to not connect directly out to, for example, Google.com, but to route all connections through the local host, as it's called. You know 127.0.0.1 is always an IP for your own machine. So the server, the proxy server running in your computer sets itself up on your own machine. And your browser makes all of its connections there. It receives this request with all the browser headings; and, for example, cookies, as well, are all available to it.

And local proxies like Proxomitron and Privoxy are able to go in and snip out things. They can blank them out. They can remove headers. They can add headers. Essentially they're very powerful, typically script-driven editors, sort of on-the-fly editors of anything going out and coming in. Remember that you make your request to it. It can edit the request and then send it out on your behalf. When it receives the reply, it comes back to it rather than to you, it's able to do any kind of filtering that you might want done. And this is why this Privoxy was originally called the Internet Junkbuster, was it would do all kinds of stripping of ads and other junk from incoming pages.

So now we have add-ons in our browsers that do that. But when you think about it, if we have different browsers, like many of us have IE and Firefox, some may be experimenting with Chrome, and some people use Opera, well, the features available on any given browser are going to be a function of its own capabilities and what plug-ins are available. Not all of the same plug-ins are available for all of these different browsers. This sort of centralizes that job in one location. And in fact, you can also have these things running on one computer in a household network and have all the browsers in the household network told to use that one computer as their gateway to the web, as their proxy, in which case you can centralize the kind of filtering and configuration, basically web page editing that you do on the fly. So that's another very powerful capability that proxies bring.

And, finally, there's a really interesting type of proxy which requires no configuration at all. And this, for example, is what many people in China are using, and in other organizations, and even, for example, in schools that are blocking Facebook access and Twitter access and so forth. All you have to do is go to a different website, where the website will proxy on your behalf. And there are, by all measures, apparently tens, if not hundreds of thousands of proxies, open proxies that are available in the ways I've talked about, where you configure your web browser to access them directly. But also increasingly popular because they require no configuration are so-called "anonymous web proxies."

So you go to one of these sites. And what you see when you go there is just field, a form field, prompting you for a URL. And so this is a website which you're visiting that is asking you where you really want to go, where you really want to visit. You enter your URL there. And that web server, acting as a proxy that requires no configuration on your part, it goes and pulls the page on your behalf. It goes and gets the page and returns it to you. What it does in the process is very clever, though. They encode the domain that you have gone to in the returning page.

So what your browser URL shows is the domain that is doing the proxying for you, followed by a long tail of gobbledygook, just cryptographic-looking noise. And any of the page assets, like scripts and CSS files and images, all of those things, those are modified by this proxy, which through your going through it has interposed itself, this proxy website modifies all the URLs, all the links on that page that comes back and all of the objects, so that your web browser then asks for those objects to finish populating the page, sort of as aliases of what they really are. Your web browser asks for those of this intermediate website, which turns around, looks up what they are, or decrypts them, and then makes the request out on the Internet. That asset comes back. And you end up seeing a web page. Also all the links, if you hover your mouse over the links, you'll see that none of them are links that used to be. They're all obscured. They're all encrypted.

What this means is that, simply by routing your traffic through this website, which is essentially rewriting your web pages, you have hidden your actual IP from the site you're actually visiting because all of the requests come from this intermediate site. And your own records, your own cache has no privacy-busting records in it. All it has are URLs of this intermediate server. All the images it caches, anything it downloads, all the links it visits, any trail, any logs that are being kept are all obscured, and they only have URLs of the intermediate server. And the way these servers have been designed, all of those links expire. They're only good briefly. So nobody coming along afterwards, looking at your browser cache, can look up those URLs and find out what they were for when you were pulling them. They're dead now. They go nowhere.

Leo: Isn't that great.

Steve: So it's really cool. And I got a big...

Leo: Why do these guys do this? Is it just, do they make money at it? Do they charge for the service?

Steve: They're free. Some of them offer, like, upsells for additional services. I got a kick out of, when I Googled just the phrase "web proxy," the first link that came up was a site called HideMyAss.com.

Leo: And it does a lot. It does anonymous email, it does port proxies, web proxies. It has a VPN. This is kind of interesting. The VPN is what they're selling.

Steve: Yes. But you can go, if you go to HideMyAss.com/proxy, or just select that on the home page, it'll take you to a page that's very lean, that simply has a place for you to fill in a URL. And if you put something in there, then hit Enter, it will take you to that site. But notice it's edited the page. There's a banner at the very top where you have some controls. You can say I want to remove cookies, I want to filter scripting, I want to do different things. So you can configure what it's bringing back to you. And there are, like, directproxyserver.com, onionproxy.com, onlineproxyservers.com. Leo, if you put in onlineproxyservers.com, and many of these are, that's a list of hundreds of these open anonymous proxy sites. And so they bill themselves as a means for allowing people who, for whatever reason, want to - they don't want to leave records of where they visit.

Now, if you control your own computer, that's probably not a problem. But maybe you're visiting someone. Or you're in a library or something. There might be an instance where you don't have the ability to, like, clean up after yourself or scrub your own trails if you need to. These services are always available. So you just route yourself through one of these places, and all of the logs, all of the content that comes back is obfuscated by these crypto tokens that have a short life. And it's certainly the case that organizations could get wise to this. For example, it's not like there's no way for your corporation to block HideMyAss.com. They could block that.

Leo: Of course they could.

Steve: In addition to LastPass, I mean, in addition to Facebook and Twitter and other things. But I think that's why there are the numbers that there are. I mean, there's hundreds of these things. And you can go to directproxyserver or onionproxy or onlineproxyservers.com and just try them until you find some that your organization or your school district or whatever organization hasn't blocked, and that allows you to get out and get to Facebook and do whatever you want to. And I'm sure that's the same approach that is being used for busting through national boundary firewalls. There's just too much available for people who are trying to police this to track down every last one of these. And they're coming and going very rapidly, so it's also a constant moving target.

It is worth remembering, though, that you need - there's some implicit trust you're placing in these anonymous proxying services because, while your computer isn't retaining a record of all of the URLs, the actual resolved locations that you've gone and you've clicked on, they're all being anonymized, essentially, that site that you go through, it knows your IP because you've connected into it. And it knows everything you actually did because you went to, well, because it was forwarding your actual requests out, after decrypting the links that you were sending it. It knows where you went.

Now, what I don't know, I haven't tried it, is if you can chain these. I don't know why you wouldn't be able to, within one, go to another. And if that works, then, let's see, the one you connect to would have your IP, but it would have obfuscated URLs from the outer one. The outer one would know where you were really going, but it would not have your IP. It would only have the IP of the first one you connected to. So, yes. If they allow you to chain - and I can imagine they might be able to detect each other. But if they allow you to chain, then it would take comparing records from both of them in order to backtrack and basically do what a single organization would have at a single point of contact, if you did chain through them.

Leo: That's what TOR does; right? That's the idea of TOR.

Steve: But TOR does it with...

Leo: More sophisticatedly.

Steve: Oh, yeah, industrial strength. As we've talked, when we discussed and delved into the detailed operation of TOR. But we've never talked about any of this before, and I thought that our listeners would find it interesting and maybe helpful and useful, if nothing else to avoid waiting for their browsers to start up.

Leo: JimmyMac3 asks an interesting question in the chatroom. How does this affect SSL pages?

Steve: Good question. Now, some of the better sites, like HideMyAss, you'll notice that on HideMyAss.com, because it is also a commercial provider, so they've got a little more technology, you can click there that you want SSL, and it will create an SSL connection between you and it. So that's a perfect example of one way to surf safely in an open WiFi. There are, in these big lists of proxy servers, they often show whether the proxy server supports SSL. And that means SSL between you and them. And that's what you want when you're in an open WiFi hotspot.

So if you used a service that looked reputable, that you felt you could trust, like HideMyAss.com, which is the first thing that comes up if you Google "web proxy," then they do offer an SSL option. What that does is it connects your connection to them to SSL, meaning that your traffic, then, as you surf through them out to the Internet to do Facebook or Google or whatever, or a service, as a better example, that does not offer the SSL protection that you want, then you're at least protected from your traffic out to them. And then it would go non-SSL from them out to the final destination.

Leo: Good to remember that.

Steve: Yeah. So that would be nice, if you were in an open WiFi situation, if these are available there.

Leo: Yeah. Steve, great subject. Very interesting. Of course, this is why these are so effective, and that's why regimes like Egypt just turn off the Internet. Because they know they can't really stop you in the long run. China's the same situation. And of course, as you said, it's a huge consequence when you turn off the Internet, and China's never going to be able to do that.

Steve: Well, and I think the logic must be, they're probably going to stop 95 percent of the people, people who don't listen to this podcast, people who aren't up on the technology, who just try to go somewhere with their computer and, oh, it doesn't work; oh, we can't get there. Well, there are always ways around that. And proxying is the oldest and longest-standing way, which is still standing today.

Leo: Steve Gibson is the man at GRC.com. That's the place to go for Steve's great program, the world's finest hard drive maintenance and recovery utility, SpinRite.

Steve: Yay.

Leo: You can buy it there, and that's Steve's bread and butter, so we encourage you to do that as a form of support, if nothing else.

Steve: And it's good for you.

Leo: It's good for you. It's healthy. He also has lots of free stuff there, including, of course, this podcast, in both 64K full quality audio and 16KB somewhat less full quality for those of you who are bandwidth impaired. He has the show notes there, and transcriptions, too, which is of course the smallest way to participate in this show. And you can read it, absolutely. GRC.com. Next week a Q&A. So if you want to ask a question about this or anything else Steve talks about, or anything that's on your mind, go to GRC.com/feedback. There's a form there you can fill out for a question.

Steve: Can we tell our listeners about the stream you've got of the TWiT studio being built?

Leo: Yeah, sure. You can watch it being built at Dropcam.com.

Steve: Slash demo, I think, isn't it?

Leo: Slash demo, yeah. I'm not sure why it's demo because it's no longer the demolition, it's actually the building. But I guess we're stuck with the URL now.

Steve: Oh, I thought it was as in demonstration.

Leo: Oh, demo Dropcam, you're right. And we're one - okay, you're right. And we're one of the ones on the left-hand side there, new TWiT studios. And thank you, Dropcam, because they're providing the bandwidth for this. We couldn't do it. And you can see what people are doing at this very moment.

Steve: So this is the new TWiT studios. It's a camera stuck up in a corner that is basically looking out over the construction of where you guys are going to be in a couple months.

Leo: Yeah, yeah. We're very excited about that. In fact, tonight I'm going over there with our designer, Roger, for his final plans and the final approval. So you're going to see a lot more action in that cam in the next few days as they start to actually construct the set.

Steve: Good, because I think it's lunchtime right now, Leo. Not much going on.

Leo: There's nothing much going on. They're just kind of quietly wandering. Hey, a program note about next week, and I haven't talked to you about this yet.

Steve: Oh, yeah, the Mac, the big March 2nd announcement.

Leo: Thank you for paying attention, yeah. It is now confirmed that March 2nd Apple has an announcement. We don't know what it is, but we presume it's iPads. It may even be new MacBooks. So that will happen during, nominally, the time that we'd record this show. So what I'd like to do with you, if you don't mind, is swap this show with MacBreak Weekly.

Steve: Perfect. So we do me on Tuesday.

Leo: Yeah. We'll do you on Tuesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern at live.twit.tv. And this Wednesday slot, which is normally your slot, 11:00 a.m. Pacific, 2:00 p.m. Eastern on Wednesdays, next Wednesday will be a special edition of MacBreak Weekly, as once again I try to incur the wrath of Apple.

Steve: Successfully, no doubt.

Leo: Well, I haven't gotten an invitation ever since that January iPad announcement. It's now the one-year anniversary of that. So I have a feeling I'm persona non grata there. But you know what, we have ways. And we will cover it live. We've got lots of great people here. And so that's exciting. So thank you. I was going to ask you. So that's okay with you.

Steve: Yup, absolutely, no problem, I'm glad we covered it.

Leo: Okay, Steve. We'll see you next Tuesday...

Steve: Thanks, Leo.

Leo: ...on Security Now!.