Roaming Authentication

Description: In this first of a two-part series, Steve and Leo discuss Steve's recent design of a secure roaming authentication solution for GRC's employees. Steve begins to describe the lightweight super-secure system he designed where even an attacker with "perfect knowledge" of an employee's logon will be unable to gain access to protected resources.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-113.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-113-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 113 for October 11, 2007: Roaming Authentication.

It's time for Security Now!, still the best science and technology podcast in the world - until this time next year, I guess. Steve Gibson, hello.

Steve Gibson: Thanks to our listeners, who helped us become the People's Choice Best Science & Technology Podcast. It was really fun, in fact, was it two weeks ago I guess that I was out at Ontario. Oh, no, it was a week ago. And it was really great. It's fun, Leo, to just sort of hang out with all of the podcast people...

Leo: It is.

Steve: ...and hobnob.

Leo: That's the best part of PME, yeah. Just hang out. And so tell me what happened. So there was the awards party.

Steve: Well, yeah. There was the awards party at the end of the first day. And I have to say I was flattered by the fact that my award - our award, I should say - was at the end of sort of a long list. Probably there were about 20 of them for - and sort of I think it was probably in the same order as they showed up on the web page. So, you know, we were the...

Leo: You were the last.

Steve: Yeah, we were down at the bottom of the page. And we were similarly, on the program, we were down toward the end. Well, I had never heard of most of the podcasts that won. And I have a feeling that most people had not heard of most of the other podcasts.

Leo: Yeah, there's a lot of podcasts.

Steve: There's a lot of podcasts. And, you know, they sort of tend to be obscure. And I think, you know, the TWiT family is a strong movement in pod land. And so here were all the people, after the exhibits and the various tracks and sessions and conferences and educational things shut down, all the people sort of hanging around the steam trays, you know, gathering for this event. And when the Science & Technology category was announced, and Security Now! was mentioned, there was a roomful of applause.

Leo: Oh, yeah. People know you, sure.

Steve: Well, and I mean, people know us, and they know this podcast, frankly, to I think a much greater degree because of the penetration and the fact that we were in our third year and, you know, blah blah blah. But anyway, it was really neat. So I was glad to do it. I don't know, though, next year it's going to be in Las Vegas. And that's...

Leo: Yeah, I heard they were moving it.

Steve: Yeah. And I'll be interested to see or to hear whether there's a strong attendance there.

Leo: I think there'd probably be a bigger attendance. It's not as easy for you, but it's a lot easier for most people. Getting to Ontario is not an easy thing to do.

Steve: Well, and of course Las Vegas subsidizes their airfare, so that the flights you get to Las Vegas are very inexpensive.

Leo: Yeah. No, I understand why he's doing it. And there's another event that I'm going to next month called Blog World that is in Las Vegas. And I think that that's - it's just a more natural place. I mean, one of the problems I had was getting to Ontario after a day in Vancouver, then having to get back up here; and it just was very difficult to schedule it, so.

Steve: Right.

Leo: Well, anyway, congratulations. I'm glad you won. And it sounds like it was a great event.

Steve: It was absolutely worth doing. And again, I thank our listeners. Actually, when I went up to the front to accept the award, that's what I said. I said, you know, I owe this to our

listeners who made this happen for us when we found out this was happening, so.

Leo: They came through fast.

Steve: They did.

Leo: Yeah. Thank you, listeners. Well, congratulations. Now, this week we're going to do something called, what is it, roaming...

Steve: Roaming authentication.

Leo: Roman authentication. Is this authentication you would get in Rome?

Steve: Well, it's interesting. It's something we've never done before. But there will be some opportunities in the next year or so for me to include our listeners in my own development of some new technologies. And we know that we spoke a couple weeks ago about GRC's eCommerce system. I sort of gave people, our listeners, a look under the covers of the way I solved problems for GRC. I'm still in this mode of sort of nailing down a bunch of loose ends. And one of the things that I needed to provide Sue and Greg, my two employees, was the ability to connect to our eCommerce system in a secure fashion when they were away from home.

Leo: So a VPN, right?

Steve: Well, no, as a matter of fact. And for a number of reasons. But this will probably be a two-part series because it's - I think it'll be interesting, and it also allows us to leverage a lot of the things that we've been talking about. You know, we tend to talk about snippets of technology here and there, how these fundamental things work. We've never had an opportunity to look at using them to solve, like, a real-world problem, which is what I have done. And in the process, I came up with some interesting solutions that'll be fun to share with our listeners. So this will be roaming authentication, which will be part one of I think it's going to be a two-part podcast, which we'll finish up week after next, after next week's Q&A.

Leo: Excellent, excellent. Before we get to that, do you have any errata?

Steve: Oh, yeah. We've got some errata. One really funky weird thing, I had to mention it because we're a show about security. Our friend Mark Thompson of AnalogX has about 60 routers. That's six zero. He's been literally collecting them from all over the place because for a project he was working on for another company...

Leo: [Indiscernible].

Steve: He was, yes, exactly, for testing various types - he was developing some, sorry to say, some Universal Plug and Play technology. It turns out that lots of routers don't do UPnP correctly. And also doing some NAT penetration work for some peer-to-peer stuff.

Leo: Oh, excellent.

Steve: So consequently he has all these routers. Well, he was wrapping up some work, and he discovered a bizarre router, which is bizarre unfortunately in a worrisome fashion.

Leo: Uh-oh.

Steve: This is a router from a company called Hawking, which I've actually heard of. In fact, I use their KVM at Level 3. It's a very nice KVM.

Leo: I've used their antennas, actually, Hawking antennas.

Steve: Yes. So anyway, this router model is the H2WR54G. Now, the hardware version is rev. A. The boot code is rev. 1.0. And the runtime is version 1.08. This will be important for anyone who has one of these routers for the following reason. Mark was working with the router and doing some characterization. He's got this cool Wi-Fi signal spectrum analyzer to show the power levels and spectrums and so forth. And he was going through, turning off the Wi-Fi on routers. And he switched off the Wi-Fi on the Hawking router using the regular administrative interface. And it didn't turn off. And he thought, okay, wait a minute, what's going on here. So he disconnected the other routers' power and made sure that this was the router he was talking to. And he turned the Wi-Fi back on. He had it set up with WPA security, and it was working normally. He disabled the Wi-Fi on the router. Well, the SSID changed to Venus...

Leo: What? By itself?

Steve: And it shut down security, but not the Wi-Fi. The router opened wide open...

Leo: Oh, that's a hack.

Steve: ...with no security. I mean, this is, you know, from the manufacturer.

Leo: Somebody got into the firmware. I mean, that's what you would do. You would make it a known SSID, and you'd turn off all security.

Steve: Well, and the other thing is that apparently two other routers, one from StarTech and one from Edimax, they look similar. Mark didn't test them specifically, but he's gotten very familiar with his 60 routers. And the Hawking router, the StarTech, and the Edimax all look the same from a management interface web browser page sort of standpoint. So I just wanted to raise a flag for any listeners who might have, again, this is the H2WR54G. This was hardware rev. A, boot code 1.0, and runtime version 1.08. I don't know that those are the latest. You'll want to, if you have this router, just be aware that you don't want to turn off the Wi-Fi even if you're not using it. Maybe, you know, you want to wrap it in aluminum foil and put it in a steel box.

Leo: Well, unplugging it would probably be enough.

Steve: Well, yeah. But you might still want to have a router, but you don't want a wireless router.

Leo: So anyway, so this got triggered by attempting to turn off the Wi-Fi. That's when it happened.

Steve: Yes. And what it did was, you know, he went back and forth, and he verified it, he made sure - what it did was it turned off security, left the Wi-Fi on, and switched the SSID from what he had programmed to Venus.

Leo: Wow. That is wild.

Steve: So that's just so bizarre that I wanted to share it with our listeners.

Leo: Yeah. Did he ever contact Hawking about it?

Steve: No. You know Mark, he's on to his next thing immediately. But, you know, he immediately phoned me and said, you're not going to believe what I just found.

Leo: That's a really weird - and this was out of the box fresh. I mean, it wasn't like he might have got it hacked by somebody.

Steve: Nope. It was out of the box.

Leo: Weird. Weird.

Steve: Next news items is there is on its way Windows XP Service Pack 3.

Leo: Yes.

Steve: It's in beta now. It's got a whole bunch of fixes, something like 1,057 or something fixes, which...

Leo: Holy cow.

Steve: ...is really welcome because I don't know if you've set up XP recently, Leo. But if you put the...

Leo: A lot of hotfixes.

Steve: Oh, my god. I mean, you put in Service Pack 2, and that brings it down to, like, 84 or 85 patches. And you've got to do it, like, five times. You've got to patch because the patches have patches. And so, you know, you don't get the patch's patched patch until you've got the patch patch in. And then it realizes, oh, there's a security problem with the patch patch. And so it's got to patch that. So that'll be so nice to have a Service Pack 3 that brings XP a lot more up to where we are today.

Leo: Paul and I will be talking about this tomorrow on Windows Weekly. But it's my sense that it's just all the hotfixes rolled up. There's nothing else in it.

Steve: Yeah, I've heard that there's a couple things that are - one blurb that seemed to be well informed said that they have backported a couple things from Vista, but nothing major.

Leo: Oh, yeah, you'll be able use the IE7 in a protected mode, I think. Yeah, I think there were a few things. I remember that.

Steve: That would be good to have.

Leo: Yeah, yeah.

Steve: Also I think they did something with their networking came back from Vista, and also some kernel mode security stuff, like the crypto package was apparently enhanced from Vista. So it's nice that the OS we actually all want to use, that is to say XP, is getting some benefit from the OS that some of us have no choice but to use, which is to say Vista.

Leo: I shiver a little bit when I hear about Vista's networking coming back to XP because probably the most common question I'm getting these days is people who have moved to Vista, and then all of a sudden all their sharing and stuff stops working because Vista does everything so differently. It has that new control panel, and it's just strange the way Vista works. And I guess it'll all work that way now.

Steve: Well, it can't. But certainly there would never be...

[Talking simultaneously]

Steve: They would never break XP by backporting something bad from Vista. I mean, Microsoft just won't do that. So...

Leo: Well, it's not that it's bad. It's actually probably good. It's more secure. But it causes all sorts of problems with...

Steve: Oh, but Leo, but if it breaks existing XP things, that would be a catastrophe.

Leo: That's not - well, now, true, yeah, right.

Steve: Okay. Lastly, our friend at VeriSign Labs, the PIP guy, Gary Krall - and for Elaine's sake that's spelled K-r-a-l-l - he dropped us a note immediately after our last episode. You may remember we did the dark side of OpenID. And the subject of his note was, "Yes, I was listening." Because of course, you know, they're an OpenID provider that we've talked about extensively that is able to do both cell phone texting verification and also to use the PayPal and VeriSign's own hardware tokens, which is very cool. And I'm going to read his letter. I invited him to be on the show, but he said, well, they don't let us get out of the lab very much. But he did want to bring these things to our attention. First of all, most importantly, he said, "We do not recycle userIDs for the exact reason you state in your podcast. If we released an OpenID identity that another user, that someone had had, then another user could masquerade as that previous user."

Leo: Right.

Steve: "Once created, the account is then frozen and never released." So this is, again, another benefit for using these guys. If it's true, and I don't know that it's true, that any OpenID providers are recycling userIDs, remember there was some anti-OpenID blogs that I ran across that claimed that some large OpenID identity providers were recycling IDs, that's a very bad idea.

Leo: Bad ID.

Steve: Very bad ID. These guys do not, you know, VeriSign Labs does not do that. Then his second point was just, "I'm curious to know if you've tried our SeatBelt product." And of course we know Leo...

Leo: I have it. I'm using it.

Steve: You have it.

Leo: Yeah.

Steve: "Because in addition to supporting our OpenID provider, we also support eight others." Which I did not know. He said two in Korea, one in Germany. He says, "The UI has been translated into Korean, German, and French for use on those systems." And he says, "If you go to a relying party, that is, someone that you are wanting to authenticate to, and you are not logged into the PIP, we detect this when the request comes into us. And when the redirect hits us, we display a very nondescript page with no external linking, et cetera, that says you are not logged in. And in order to prevent phishing attacks, we require that the user must be first signed into the PIP before they sign into an OpenID-relying party." And he said, "To Leo's point, this allows us to ensure that the user has logged in securely with us for, in addition to having SSL throughout the site, we're also green bar-enabled," as Gary put it, supporting their extended verification certificate. And he says, "Now, how does SeatBelt come into play? When you have it installed, we'll detect if you're signed into the PIP or not. When you bring focus to a text-edit field on an OpenID-relying party site, we bring up a blind which tells the user that they're not logged in, and clicking on the button will cause them to be taken to the log-in page

of the PIP. Once the user has signed in, which if they have their token bound to gives them even higher level of authentication, we'll redirect the browser back to the relying party site, and we'll populate the OpenID URL of the user into the text edit as a form-filling function." Sort of automatically fills that in for you. So...

Leo: Now, if you have SeatBelt installed, which is a Firefox extension, you don't even have to go through that process. I just went to a participating site. It sees the Firefox plug-in and says, do you want to use OpenID? I see you're leolaporte.verisign.com. I click okay, I'm logged in.

Steve: Yes.

Leo: And that's the beauty of SeatBelt. It's just automatic. Once you've logged into SeatBelt.

Steve: Yes. And so I wanted to make the point that this really does solve a lot of the phishing concerns. Certainly everything is done with SSL, and these guys are never recycling OpenID userIDs. So, you know, those concerns that we brought up from looking at the dark side of OpenID have been dealt with by the VeriSign Labs guys.

Leo: Excellent. Excellent. That's great.

Steve: And I have one fun and interesting new twist on some SpinRite story from a listener, Andrew Baker. His subject was a little premature. He said "SpinRite Halloween Success Story."

Leo: Halloween? Not yet.

Steve: He said, "Hi, Steve. I have an interesting SpinRite story. I live right around the corner from you in Lake Forest, California. I have my own eBay consignment business, and I keep all my inventory at a local storage unit. I was about to throw out some trash, and the storage people stopped me to examine the trash for any electronics because someone earlier threw away 'three old computers.'"

Leo: Wow.

Steve: "In a bit of a curious state, I asked if I could see them because, if they worked, I could find them a home and would be happy to give them a finder's fee. They proceeded to show me three Pentium 4 shuttle-like PC machines, all with sticky notes on them saying 'bad drive.'"

Leo: Oh, man. I see where this is going.

Steve: He says, "This is where the interesting part comes in. After paying the storage unit employee 50 bucks for all three questionable machines, I took them home and whipped out SpinRite. SpinRite fixed the drives and recovered the whole machine in every case," he says, "although I was locked out without a Windows admin password. So I started typing in random common passwords. After the second try, I got in. It was their username."

Leo: Oh, dear.

Steve: "I thought it was comical..

Leo: Whoever was running this is an idiot.

Steve: He said, "I thought it was comical that SpinRite helped me get the discarded computer working, and how then I was able to circumvent the password so easily. I'm an honest person, so I just wiped the drive. In any case, it shows how SpinRite works well. So don't let your non-working machine get into the wrong hands."

Leo: There you go. Good advice. Wow. And they threw it out.

Steve: Yup. They just said, well, these are bad, so I'm going to throw them away. And it turns out, I mean, just run SpinRite on them, and not only are they back, but of course all their data was then available. So it was good that Andrew is an honest guy and just wiped the drives.

Leo: Wow. All right. You've got - here's the problem. Here's the situation. You've got employees offsite. And they need to log into the system.

Steve: Yes, exactly.

Leo: Now, normally a business will set up a VPN to do that. Right?

Steve: Well, yes. A lot of businesses will use a VPN when they've got users roaming, and they want to give them access to their corporate network.

Leo: It's kind of a heavyweight solution.

Steve: Well, and it turns out it's problematical. I was at dinner a couple weeks ago, and it just happened that, out of the blue, two of the gals that were part of this group commented that they've been unable to connect to their corporate networks through their VPN. One happened to be at a hospital, and the hospital's network was hostile, apparently, to her doing that. And the other one was just, you know, somewhere where they were unable to connect. And so I wanted a robust solution. I've seen problems, for example, even with OpenID. OpenID is a routing-based approach. And if the network you're in happens to have the same address space as the network you want to connect to, there's no way for OpenID to know whether the packets are intending to stay local or be routed over the VPN.

So it turns out, I mean, I've got some experience with OpenID, and I run across all kinds of problems where, nice as it is, there are situations which it's unable to deal with. But mostly I didn't need to give Greg and Sue access to our corporate network. And in fact I'd rather not from a security standpoint. That's almost too much power because, well, again, from a security standpoint you never want to enable more access than is necessary. And, for example, the way our system operates, web access is all they need. The way we're set up now is that our corporate network knows my network range where I am in my office at home. And it also

knows Sue's router IP and Greg's router IP. Which even though they use DHCP, and technically the IP can change, if you don't ever turn off your router, that router tends to hold the IP from your provider.

So the way our security functions now is very strong. Only connections from my network or Sue's IP or Greg's IP have the ability to bring up any of the management pages of our eCommerce system. So literally, we know from having talked about spoofing in the past, there's no way to spoof an IP of a TCP connection because you have to have a valid return address in order to accomplish this three-way handshaking in order to create a TCP connection. So given a situation where you've got fixed IPs, it's pretty easy to create strong security. You tell your firewall, allow connections in from, you know, TCP connections from this IP.

Now, when their IPs change, for example, sometimes Greg will turn off his router to let it cool off or whatever, or if he's away for a long time or there's a power outage or something, it is possible to lose the IP. But they just give me a call, or they send me a piece of email, and their email will contain the IP of their client at this time, which will be their router's IP. And I just edit a registry entry in our server, in our corporate server array, and they're reauthorized. So I've solved the problem in a strong way, as long as they stay home. But Sue would like to take a vacation every so often.

Leo: Yeah, right.

Steve: And she has a need, when she's away from home, like she's on vacation, to have access to the management interface on our eCommerce system. For example, customers often will say, hey, I'm away from home. I'm on the East Coast; I live on the West Coast. I desperately need access to my copy of SpinRite, but I don't have it. I didn't bring it with me, and I don't have my receipt that has our transaction ID. Could you look it up for me? So, and we have all kinds of ability to do that, but only if Sue has access to the management interface for our eCommerce system, which is tied, currently tied to her IP. So inherently, if she's out roaming around, she has no such access.

So I did look at, okay, you know, how about a VPN to her base station? That is, allow a VPN connection from wherever she is to her machine at home. Well, that requires that she keep the machine at home on during a week's vacation, for example. And that's not an insurmountable problem. But again, we still have the problem with current VPN technology being problematical. And just to clarify, the reason that could work is if she were to VPN to home, then go from there back out onto the Internet to GRC's network. GRC would see this connection as originating from her residential IP and would then authenticate. So, I mean, it's sort of the solution. And I looked at it for a while, but I decided, you know, that's not what I want to do. I want to come up with something that is absolutely robust, and at the same time absolutely secure.

So the next thing I looked at was client certificates. That's a technology, again, that we've talked about before. We've talked about the normal case where a browser has sort of generic certificates that it uses for establishing SSL encrypted connections to servers. And we've talked about this, the large and the growing number, in fact sort of a worrisome number, of certificate authorities that are now being trusted by our browsers. And the idea there is that the web server uses a certificate to authenticate itself to the browser. Well, it's possible to give the client, that is, the web browser, a certificate, and sort of have it present its certificate to the web server as part of its connection. So I looked at that for a couple days. And I decided, you know, that's got problems, too. It's a bit of a pain to administer. Maybe if we were a big, huge organization, and we wanted to have a certificate server that was going to be managing certificates and have people coming and going, I could see that it would make sense. But it just sort of seemed like overkill for us. So I was looking for something that would be simpler to use.

So I fell back to the idea of using some cookies. Now, cookies are generally considered not

secure, and for one reason is that they're sort of a stock-in-trade with browsers and servers. It turns out, though, that it is quite possible to use cookies in a secure fashion. There's an argument that you give to a cookie which is secure. It was originally defined by Netscape, and all browsers honor this. The idea is that, if you flag a cookie with a secure tag, the browser will only offer it to the matching server if the connection has been secured, that is, over an SSL connection.

So just to remind our listeners how cookies function, the idea is that a web server is able to issue a browser a cookie. And it does so by - essentially to the web server's domain name, such that any time the web browser is asking for resources from that server, if it has matching cookies, any cookies that match the domain of the server, the browser offers them back just as part of its identification process. So it allows the server to track the user as they move through the website by basically giving them a tag so that they're able to know that this is the person on the site.

So my thinking was, okay, falling back to this notion of secure authentication, I wanted to essentially have a multifactor authentication solution where we would have something that the user had and something that the user knew. So I wanted to basically authenticate the laptop that Sue or Greg, or myself for that matter, would be roaming with. And so the way to do that was to give the laptop a cookie which could not be sniffed. I was originally concerned about, well, what if they had a nonsecure connection to GRC, like looking at non-secured pages, would that disclose the cookie? Because I wanted to absolutely make this non-sniffable for security's sake. Well, that's where tagging the cookie with the secure flag comes in. No browser will disclose a secure tagged cookie. Which is very handy for this sort of purpose.

Leo: Well, wait a minute. It has to disclose it to somebody. It'll disclose it only to the originating site?

Steve: Well, it won't - you're right. I'm sorry, I didn't explain that well. It will never send the cookie back to the web server, even the issuing web server, unless it's over an SSL connection.

Leo: Oh, okay, I got it. Yeah, because the normal cookie policy is only send cookies back to the originating cookie setter.

Steve: Exactly.

Leo: So this adds the SSL to it.

Steve: Yes, exactly. If the server tags the cookie as secure, then that informs the browser never to send a cookie back unless it's over a secure connection with that server. So essentially what this makes the cookie is non-snoopable, non-sniffable. You do not need to worry, then, that somebody who was monitoring your traffic and sniffing the traffic would ever be able to get access to the cookie. SSL is a good, strong, man-in-the-middle snooping-proof technology when, you know, when it's being used correctly. And we've talked about SSL proxying and the need for a proxy to have a certificate on the client in order to have permission to decrypt the session and then reencrypt it. So, you know, of course none of that would be in place. So it means that just using the cookie and flagging it as secure gives us a strong means for authenticating that machine.

So in terms of the mechanics of this, the way I have it set up is - so the question is, how do you plant the cookie on this laptop? The only way that the laptop can receive this cookie is if it is at the recognized management IP address. So, for example, Sue has to have her laptop at

home. And when she uses the laptop to go to these secure management interface pages at GRC, the server recognizes that it is getting a connection from her home IP, and that the request for the management interface did not have this cookie present. So if the cookie's not present, it intercepts the request with a screen that comes up and says, hey, wait a second, this machine has neither been authorized nor deauthorized for future roaming access. Meaning that we haven't told GRC one way or the other whether this machine should have roaming access in the future. Meaning that this cookie's not present either saying yea or nay to, if this machine is not at the management IP in the future, should it be a candidate for access to our network?

So the idea is that, for example, if a visitor brought their machine to one of my employees' homes, and for some reason Sue used it to access GRC, it would see that this machine was contacting GRC from an authorized IP, that is, one of our known network IPs, which normally gives authorization, but that this machine does not have this special GRC roaming permission cookie. So an intercept page comes up and says, do you want this machine to have future roaming access? Well, if this was not her laptop she would say no. That gives it a cookie that is similarly a secure version of this cookie whose value carried in the cookie says this machine is not - if this machine ever attempts to access GRC from another IP outside the authenticated IP, don't give it access. So that's just a clean way, basically, of identifying any machines which are ever at the authenticated IP address, to identify whether or not they should be able to have roaming access if they are subsequently not at an authenticated IP.

Leo: So it has to be a machine that you've seen before and from an IP address you've seen before.

Steve: Well, actually, yes. A machine you've seen before, which once was at - it was physically located at one of our authenticated IPs, and the person said, that is, Greg or Sue or I, yes, I want to give this machine future roaming access.

Leo: Ah, okay.

Steve: So that allows the machine to carry this cookie, which is flagged as secure, which will never be exposed. And any time they're out roaming around then and establish a secure connection to GRC, that cookie resident on their machine will be sent back as part of the query to allow them to access GRC.

Leo: And you can say that this cookie will never expire?

Steve: Yes. In fact, it is, it is a non-expiring cookie. It is - I think I expire a non-expiring cookie in, like, the year 2047 or something.

Leo: By then, who cares?

Steve: Exactly. We'll have a whole different authentication technology by then. So, okay. So we've solved part one of this problem. We've basically, you know, the three of us have laptops. By using our laptop at our home base and going to the secure pages on GRC, we are presented with a dialogue saying, do you want to allow this machine in the future, if it's roaming outside of these authorized IP ranges, to have secure access to GRC? We say yes. That gives it a cookie, a non-expiring cookie, tagged as secure. So essentially this identifies a few machines that, if they connect to GRC not from an authorized IP range, that they're candidates for having

access to the privileged management interface.

Now the problem is, how do I prove that this is Sue or Greg or myself that is using the laptop? Because essentially we've verified that this machine is authenticated. But again, you know, I want this to be as secure as possible. How do I prove that this is Sue or Greg or myself? Well, the next thing that you would expect is to use a password. And certainly many people have gone to websites where you're prompted with a dialogue box that pops up and says, you know, please enter your username and password for access to this site. I could do that, except that we have the problem of keyboard sniffing. We know that anything that they type into the keyboard could be captured. Now, we certainly hope that our machines are virus-free, that there's no malware or trojan on them. But we have seen situations, for example, in fact I've told a story in the past where Sue got the web searching, the CoolWebSearch, that's...

Leo: Ugh, yeah.

Steve: Yes, yes, it was CoolWebSearch on her machine. You know, she innocently went to some website that installed it on her version of IE, and it created a huge catastrophe. I mean, not that it was any big problem for us, and we were able to get it off. But in removing it, it broke her connectivity. And, I mean, it was your typical malware nightmare scenario. So we do know that it's theoretically possible for something bad to get onto our computers. And we also know that something that's going to log keystrokes will be paying attention when it's logging keystrokes to a secure site. It's like, whoa, you know, this is high value potentially. So let's log the keystrokes and the URL and go email them off to somewhere in Russia, Lithuania, or China or somewhere. And we'll let the evil ones who receive this figure out what they want to do about it. So there's no way that I'm going to allow anyone to be vulnerable to known problems. And certainly keystroke logging is a known problem.

So then I was thinking, okay, well, we've been talking about authentication a lot. We've been talking about the VeriSign and the PayPal dongles and using that. I've never managed to get access yet to the API from VeriSign. And I sort of, for GRC's purposes, I...

Leo: That's kind of overkill anyway; right?

Steve: Well, maybe. I shy away from depending upon any third party. I have an acronym that I've used. I shared it with you, Leo. TNO is Trust No One.

Leo: Yeah, right.

Steve: And I like it because I just - I didn't want to rely on any third party for any reason. I mean, these are the keys to the kingdom. These are GRC's most precious jewels. And I just, you know, I want control over this. So I thought, okay, I need to come up with something where even if everything bad that could possibly happen is happening, we are still secure. Meaning that, even if there was keystroke logging software installed on one of these machines, that monitoring the keystrokes is not a problem. So I thought, okay. That means some sort of a one-time something or other.

So my first idea was to present us, the three of us, any of the three of us, when we want to establish a session to this management interface, to present us with some sort of a puzzle, something that we know how to solve that involves data that we're given where, like, for example, we're given a jumble of numbers and letters. And there's a rule for how we take these and reorder them or sort them or add up the digits and do the alphabetic things in reverse order or something. I mean, the idea being something we know that, when presented with a

puzzle, we uniquely have the ability to solve it.

And I thought, well, that way the server would present us with a puzzle. And we would demonstrate we have the unique GRC knowledge of how to solve this puzzle by entering in the result. We would never be given the same puzzle twice, so we would never be entering the same result. So even if something were watching us, then repeating that action, which of course is the danger with a keystroke logger, repeating that action would not work a second time. And my idea was that, after we once solved the puzzle, then we would be given a session key, that is, a session cookie, essentially. So we have the static cookie which - and it's the presence of the static cookie saying this machine has been given permission to have roaming - potentially have roaming access. And the presence of that static cookie would then present the puzzle. If solved, we would then get a session cookie which, as long as the web browser is open or, for example, maybe as long we do something within an hour, that keeps the session alive. Otherwise it expires. And if we try to use again the GRC management interface, we'd be presented with another puzzle to verify that, you know, we are still a GRC employee.

Well, I thought about that for days. And I could not come up with anything that worked for me. I could not come up with any sort of a puzzle where, if you saw the puzzle, and you saw the result, it wasn't obvious how you got there. That is, again, you have to presume perfect knowledge on the part of the attacker, that is, that the attacker will have something that is able to capture the web page and capture the keystrokes that result, and basically have the same information that we have. And, you know, looking at this and doodling on paper for a couple of days, I could not come up with something where, I mean, I had to admit to myself, okay, if someone said what's the relationship between these two things, especially if you had several instances of that, if this malware was installed, and it was keeping track of this, where you look at three puzzles and the respective answers, where you couldn't easily yourself reverse engineer what the secret was.

Leo: Well, I think you wouldn't get a job at Microsoft.

Steve: How so?

Leo: Well, because they do those puzzles. That's what they do in the interview. They sit down and say, how much water is there in the world, in gallons? Actually they don't care what the answer is. They want to see your process. So they probably would watch your process right up to this point and say, you're great, but then you couldn't come up with the last piece of the puzzle.

Steve: Well, I did, Leo.

Leo: Oh, you see, you would get the job then.

Steve: I came up with something.

Leo: But you're not going to share it with us, are you.

Steve: No. We're here 50 minutes in, and...

Leo: I knew it.

Steve: I'm going to leave our listeners with a cliffhanger. I have something that I really like a lot, which is it's very cool, and it's a slick system. I've even made the - I've implemented it. I've written the software. I have a whole crypto system. And I'm making it all available to the industry for free.

Leo: But not yet.

Steve: In two weeks we're going to find out about this cool solution I came up with for authentication, which is practicable, which is usable for us. I'm going to use it in a future product of GRC's. And I'm hoping that making it available and open and explaining how it works to our audience will encourage other people to use it. We will be talking about it in two weeks. And we'll let our listeners wait to learn about it.

Leo: Aren't you coy. See, I knew this all along. I was just leading you on. If you are a Security Now! fan, you can go back in time through all 112 shows, transcripts, notes, and a lot more at Steve's site, GRC.com/securitynow. He also has 16KB versions for the bandwidth impaired, makes it easy to download. And our license encourages you to distribute this, send it to your friends, make sure everybody hears and gets the important security news that we cover in this show each and every week. The best science and technology podcast in the world.

Steve: [Trumpeting].

Leo: [Trumpeting]. GRC.com is also a good site to go to for [SpinRite], Steve's fantastic hard drive recovery and maintenance utility; for all of his free stuff. He gives away so much free security stuff, including ShieldsUP!. Did I say ShieldsUP!? I meant SpinRite in the last thing. Now I'm talking about ShieldsUP!, which is a tester for your firewall or your router. He also does, oh, so many other great programs. And it sounds like he's got something new up his sleeve. GRC.com.

Steve: I think our listeners are really going to get a kick out of this. I came up with a very nice solution that is, you know, it answers my need absolutely, provides absolute, I mean, flawless security. It allows somebody to literally be looking over the shoulder of my employees and myself, to be monitoring our connection, and it won't help them at all.

Leo: See, I like what - I enjoy following your thought process because you really do think out of the box. And while most people would just cobble together something existing, you like to do it from scratch. And so there's a lot of interesting ideas in there about ways to do these things. You really think about it, which is really neat. So we'll find out more about that. Actually next week we've got a mailbag episode for 114. And then for 115 we will find out more about Roman [sic] authentication.

Steve: Right-o.

Leo: I like it. Thank you, Steve. We'll see you next time.