Rootkits

Description: This week we explain "rootkit technology." We examine what rootkits are, why they have suddenly become a problem, and how that problem is rapidly growing in severity. We also discuss their detection and removal and point listeners to some very effective free rootkit detection solutions.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-009.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-009-lq.mp3

Leo Laporte: This is Security Now! with Steve Gibson, Episode 9, for October 13, 2005: Rootkits.

Steve Gibson is here. We are in Toronto, preparing to record another episode, or two or three or four, of Call for Help. And today we're going to talk about rootkits. What is a rootkit?

Steve Gibson: Well, you know, in prior episodes of Security Now! we've used the term. We've sort of bumped into them tangentially. And I thought we ought to just, you know, take some time and really explain what that is because it's no longer just a theoretical problem. It's really becoming something that end-users need to be aware of.

Leo: And that's not good.

Steve: The term "root" is the original term used for, like, the maximum privileged account on a UNIX machine. UNIX, of course, beat Windows to the Internet...

Leo: Right.

Steve: ...by, I don't know, a decade or so. You know, it was the foundation of the Internet. And so when hackers wanted to get into and compromise a UNIX machine - because UNIX administrators are typically very savvy, you know, they're not Mom and Dad, they're like, I mean, they knew what was going on. They couldn't just install files and run processes because they would be spotted immediately. So they realized they had to go stealth. So rootkit technology refers to - it's a stealthing approach. Anything that stealths a process or files or, in the case of Windows, registry entries, basically - and even the appearance of the process in a list of processes, so that stuff can be loaded on your hard drive and running in the system. And no matter what you do, you can't see it.

Leo: It's undetectable.

Steve: It's virtually undetectable. We'll talk here before we're done about new approaches that will detect rootkits. But unless you take extraordinary measures, you can't see them.

Leo: It seems like that's a pretty big flaw in an operating system, that a process should be able to do

that. I mean, can't they just say nobody's allowed to be this stealthy?

Steve: You know, the problem is, no OS we have was designed from the beginning assuming mistrust. One of the other things we're going to talk about in the future is the fundamental problem with security being assuming trust, and dealing with problems rather than assuming mistrust, and then explicitly allowing. Our whole security model is upside down.

Leo: It's too bad. I mean, it'd be nice to say everybody's beneficial, nobody's being mean. But in fact we know now that, no matter what you're designing, somebody's going to try to attack it.

Steve: Well, and imagine having a system that is fundamentally hardened against attack, instead of all of our systems being fundamentally open to attack.

Leo: And patched. Even UNIX, though, originally was not designed to be secure.

Steve: Well, it had the idea of, like, the root was the way you had, you know, godlike powers over the system. And then you were supposed to have normal users that couldn't mess around with the operating system.

Leo: Even from the beginning, UNIX had permissions and access control and things like that.

Steve: Well, and so did Windows, actually. I mean, you know, the original NT was a secure operating system that had the ability to...

Leo: Well, NT. But never Windows 95, 3.1, 98.

Steve: That's a very good point.

Leo: It was inherently insecure. And in fact I think that's part of the problem XP has is that, in order to maintain compatibility with older 98 programs, they can't really have secure policies.

Steve: Right.

Leo: They have to allow programs to do things they shouldn't.

Steve: Exactly.

Leo: Yeah, yeah. So who's using rootkits now besides mean little hacker kids?

Steve: Well, what's happened is, in this ongoing escalation of spyware and anti-spyware, as people are becoming more aware of spyware - and, for example, the antivirus companies are now beginning to scan for malware, and of course we've talked about spyware itself, - Ad-Aware and Spybot Search & Destroy. There are now spyware scanners that attempt to find this stuff on your system. So what's happening is this rootkit technology has come down from on high, and it's well-known now there are sites like rootkit.com on the Internet that have the source code for this stuff.

Leo: Rootkits R Us.

Steve: And remember Phrack magazine, Phrack.org, P-h-r-a-c-k, you know, they have a magazine that is published every so often. I think they're, like, up to issue 63 over the last...

Leo: Yeah, not anymore though. They stopped, but...

Steve: Yeah, well, they did shut down finally. But the last issue is out, and it's full of source code showing how you can do this. So what's happening is, the spyware and commercialware and other sorts of malware are now incorporating this rootkit technology to go stealth. Once they load themselves on your system, they disappear.

Leo: When you say "disappear," you mean if I hit Control-Alt-Delete, I don't see them in a list of processes? If I'm running an intrusion detection system, I can't see them there? The antivirus and the firewalls don't see them? They're totally invisible.

Steve: It's freaky. I mean, when you play with this stuff, it gives you a sort of a chill because you know it's on the hardware, and you know where it is. You do a directory listing, and it doesn't show. Now, you know, we have hidden files, for example, where we're able to set the attributes to a file to "hidden" or "archive" or "read-only," you know, at that level. But this is a level where you can't unhide these files. Your operating system will not show them to you.

Leo: How do they do that?

Steve: What happens is, they essentially modify the way the OS itself works. They're compromising the operating system kernel. You know, in operating system terminology we have the notion of a kernel, which is the OS core. And then you've got applications which run as sort of clients of that operating system. So a program you're running, you know, Corel Draw or Outlook or whatever, that's a client of the operating system. Well, so are the spyware scanners. So when you're running even a spyware scanner, it's saying to the operating system - in fact, for example, there are two API calls that's "find first file" and "find next file." So if you ever want to, like, do a directory listing, you'll say "find first file *.*," and it gives you the first file. And then you successively call "find next," "find next," "find next," until it returns no more files. That's all there is to it. So that's - so anything that's scanning your system is basically doing that.

Well, imagine if something altered the way the "find first" and "find next" operated, so that it was intercepting the response back to you, out of the operating system, back to any application that was asking, so that if it was about to report one of its own files, it would call - it would say, whoops, and call "find next" again on your behalf, skipping over that file. Suddenly any program running on the operating system will not see any of those stealthed, rootkitted files. They just disappear.

Leo: Do they have to kind of rewrite the operating system kernel? They replace the kernel to do this?

Steve: Actually, no. Turns out that the way the OS is linked together, there's a table of jumps at the end of every program which is called the IAT, the Interrupt...

Leo: Access?

Steve: ...Access Table, yeah. And so all calls vector through that, look up the address in the operating system, and then go off to there. All you have to do is change that address to your own code, and so the call goes to you instead of to the operating system. And then from there it goes down into the operating system. So, I mean...

Leo: You might say, why did they make that so easy? But there are legitimate reasons why you want to have an IAT, yeah?

Steve: Well, for example, I've done that myself. On my own code, I want to make sure that I don't overwrite memory. So I've intercepted the memory allocation and freeing of my own software so that I put guard bands in front and after blocks that I allocate. And whenever I allocate memory, I fill them with a pattern. And whenever I deallocate, I make sure that that pattern is still present. So it's very cool for me. I was able to add a feature to the operating system that it didn't have for my own software by doing this. Well, imagine that something else comes along and starts adding malicious features to the operating system. It's completely transparent to my software that I'm checking my own memory allocation. Similarly, it would be completely transparent to some other software that something is altering what the operating system is doing.

Leo: Can't a tool, a security tool, look at that jump table and make sure that it's intact, or somehow maybe write its own directory scanning routines, somehow avoid this?

Steve: The problem is that any tool like that is using the operating system even to do that.

Leo: They just have to prevent me from doing that, and...

Steve: Once trust is broken, it's broken. So once something gets into your system, you don't know what it's doing and what happened.

Leo: And you may not even know it's there because there's no way of finding out.

Steve: Well, that's...

Leo: Because you have to use the operating system.

Steve: Exactly. That's the biggest problem. The only thing that you would see would be, like, behavior that would give it away. For example, there's no way it can keep the lights from blinking on your router when it's sending data out.

Leo: Right.

Steve: And there's no way that, for example, if you had another computer sniffing the traffic, that it could hide that. Now, the problem is, it could encrypt it so it wouldn't know what it was. And, you know, Windows is so busy talking to the world now, it's all, any new updates today, what's going on today, I mean, you know, who knows what's going on with it most of the time?

Leo: Is it possible to write a secure operating system?

Steve: Absolutely theoretically possible. The problem is that, the way OSes developed, they developed in an environment sort of of academia, in the case of UNIX. It's, you know, remember that these were $100,000 systems. Nobody was going to be carrying one in their pocket. They weren't going to go, you know, down to Radio Shack and take one home.

Leo: And there weren't USB keys. There weren't, I mean, yeah.

Steve: Yeah, people literally wore white coats.

Leo: Yeah, to go in there.

Steve: And they had to - or a jacket because they were in air-conditioned, elevated floors. I mean, so there was no awareness or presumption that you'd have 13-year-old script kiddies that were, you know, talking about this at lunch in high school and coming home and trying to outcode each other to see who could hack, you know, their Christmas present from the year before.

Leo: So it's pretty widely known then that, I mean, the kids are out there. They know about rootkits. They know how to write them. They know how to use them.

Steve: Actually, it's - now they brag about "getting root" on this and "getting root" on that. That's the term for compromising and getting full, godlike power on someone else's machine.

Leo: I guess, you know, the fact that you can figure this stuff out in Windows shows that some people are pretty determined. I guess it's not so hard at all with an open source operating system. I mean, Linux would be fairly easy to find these hooks and...

Steve: Actually, that's one place you could argue, and security people do, that full disclosure is a problem. There are several techniques that we'll talk about here in a minute that are successful in finding current rootkit technology because they take advantage of still undocumented stuff in Windows. Now, we know it's only a matter of time until that information gets loose. So you might say, well, using that gives you a false sense of security. But we now that the higher the fence is, the fewer people are going to take the trouble or have the ability to climb over it.

Leo: But that's the old saying, "security through obscurity." I mean, it's certainly nothing you'd want to rely on. It's not a bad thing to have, but - but so you're saying that open source operating systems are perhaps more hackable, at least in this way.

Steve: I would say they are more hackable because they are fully known. And, now, people who want to argue against that, and I could take that position, too, it's like, well, security's bad, full disclosure's good...

Leo: At least you're going to write something secure from the get-go.

Steve: Well, the other thing you're going to do is you're going to assume the hacker has the same information you do.

Leo: That's right. That's right.

Steve: And that's really what you want.

Leo: You can't count on them being ignorant and in the dark.

Steve: Exactly.

Leo: Who is using rootkits?

Steve: That's the real problem, and that's why it's worth us talking about this for our regular, you know, interested PC user, is that rootkit technology has found its way into downloadable spyware and malware, that is part of freeware, that is part of file sharing systems, that is stuff you can - you could have it on your system right now...

Leo: And have no way of knowing.

Steve: You would not know. And none of the anti-spyware technology, Ad-Aware or Spybot Search & Destroy, those tools are not yet doing anti-rootkit scanning. Now, Microsoft's own malicious software removal tool has just started doing that because the research group in Microsoft - remember that we, in fact our very first podcast we talked about the Strider HoneyMonkeys. There is a Strider GhostBuster project. The idea is that Microsoft uses the term "ghosting" as disappearing from all detection in the system. So they have a GhostBuster as their technology which they're working on to begin to deal with this. But getting back to your question, spyware is now incorporating this really hard-to-detect-and-remove technology.

Leo: You don't want to name any names?

Steve: I know that CoolWebSearch is doing it.

Leo: Really.

Steve: Yes.

Leo: I mean, there's no business model that involves hacking your customers' computers, or is there?

Steve: Anybody who knows about CoolWebSearch and who has gotten their system infected...

Leo: Nasty.

Steve: It's nasty. Gotten infected with it over and over and over. I mean, it is absolutely detected by any anti-spyware tool because it's so prevalent.

Leo: You're making yourself an enemy.

Steve: Yes. Yes. So it's got to be that the people who do CoolWebSearch have just decided, look, we know the world hates us, we're just going to try to get in there anyway. We're going to give them this CoolWebSearch whether they want it or not, and not going to let them take it out.

Leo: And we're going to use rootkit technology to do it.

Steve: And we're going to use rootkit technology to prevent ourselves from being removed.

Leo: You give me some hope, though. If Microsoft can come up with a way to detect these rootkits, they're not completely undetectable.

Steve: Well, there are two approaches in general. There is in-system detection and out-of-system detection. The fundamental problem with in-system detection, meaning that you're running your detector on the system that you're trying to test, the fundamental problem is you know nothing about an infected system.

Leo: It's been compromised. You can't trust anything it tells you.

Steve: Exactly. Once compromised, anything could have been done. You no longer can trust any operating

system call. For example, there is one rootkit called the FU rootkit, which removes itself from the list of processes. There's, I mean, down in the kernel, internal kernel data structures, down in protected kernel memory, there's a list of everything running. That's what Task Manager shows you. That's what Tlist shows you. I mean, that's what everything uses. Well, literally, it's what's called a "linked list." It's a series of entries, each one pointing to the next. This thing takes the entry pointing to it and points it to the one it's pointing to, meaning that it unlinks itself from the list. Well, that's a bad thing to do because now you can't get rid of it. But that's what it wants. It wants to be undeletable, to disappear from any listing. It's still running. It's got threads. It's able to do stuff. But it's sort of just disappeared from the operating system.

Leo: So how does Microsoft get rid of these things?

Steve: Well...

Leo: You said there's internal you can't trust. So is there an external way to monitor this stuff?

Steve: Well, there are the fundamental problems of internal and external. Now, the guys at Sysinternals.com - and we're going to have a bunch of links for people to download this stuff, both on your show notes and on mine. There is a RootkitRevealer, it's called, at Sysinternals. And they've done something that is very clever. And actually they say it's patent-pending technology, but it's exactly what Microsoft is doing with their GhostBuster, so I'm not sure who got there first. The Sysinternals guys, they know the FAT file system and the NTFS file system backwards and forwards. What they have done is they've built a little file system technology, basically a little mini OS of their own, and it's free. This RootkitRevealer is free. And I hope everybody who's listening to this will grab it. You just want to run it on your system. It will probably find nothing. But if it finds something, that's going to get your attention.

The idea is that any rootkit is all about hiding itself. So what this thing does is this scans the operating system from the normal API; and it also, because it knows how the FAT file system works and the NTFS file system works, it also scans it itself, down at the lowest level, physical disk access level. So it's doing a scan of what the OS is telling you, and it's doing - basically it's its own little operating system, scanning individual hard disk sectors, interpreting the master file table, interpreting the file allocation table, and all of the file and directory entries. So, and what it does is it compares them. Because if there's no rootkit installed, then the operating system...

Leo: Will match exactly.

Steve: It would match exactly.

Leo: All right. So any discrepancy means that there is a file on there that's not being exposed to the operating system.

Steve: Exactly.

Leo: Something hidden.

Steve: Well, and what's so cool about this is it's universal. Now hiding yourself is a bad thing to do because it exposes you. Any change you make to the operating system in terms of files, it will find.

Leo: And pretty much anything has to at least do that, has to save itself somewhere.

Steve: Well, and many things want - the other stealthing technology is to stick yourself in the registry. The question is, how is this thing going to get started when you reboot? These rootkits want to survive a reboot. So they've got to stick themselves in all - and we know Windows is all full of these, you know, runonce, runservice, rundll, appnt, there's all kinds of ways these things can start. Well, there's a comprehensive list

of them. They're in the registry. So a rootkit has to have logged itself in to the registry in order to get itself started at boot. So the other thing the Sysinternals guys have done, they absolutely know what's called the "hive structure." The "hive" is the on-disk storage for the registry. So, similarly, they do a registry scan of all the known ways that anything could start itself in the operating system. Then they read the raw hive file and parse it in order to compare it with what the registry scan showed. And again, anything, any discrepancy, there should never be a discrepancy. Any discrepancy says something is starting up and is taking extreme measures to hide itself from us.

Leo: So the Sysinternals stuff doesn't use kernel calls at all. It never uses the operating system call at all for its own internal look.

Steve: Well, it does. It uses the operating system call like any application would.

Leo: And then...

Steve: And then a low physical...

Leo: And that couldn't be changed by a rootkit.

Steve: Well, that's the problem, is...

Leo: I knew there was a problem.

Steve: That's the problem, is, okay, in order to do that, the rootkit would have to intercept the lowest level hard disk call and be smart enough to edit individual hard disk sectors.

Leo: That's pretty hard to do.

Steve: That's the point. This really...

Leo: You'd have to know exactly what sector you're on and...

Steve: This really raises the bar. The rootkit would then have to also have a full little operating system in it in order to be able to lie about the contents of physical sectors. So it is way harder. And at this point, the free, free to download from Sysinternals, it's called RootkitRevealer - people could just Google it if they want to.

Leo: I'm doing it tonight.

Steve: Yes.

Leo: I'm running right out there.

Steve: It finds every known rootkit. Because no rootkits are yet smart enough, I mean capable enough, to fool it. But get this. This is really cool. What rootkits did start doing is when...

Leo: They'd look for RootkitRevealer is what they'd do.

Steve: And not lie to it.

Leo: Tell it the truth.

Steve: Yes. Isn't that neat?

Leo: So there's no discrepancy.

Steve: Right. So, oh, I just love that. So what happened was, after RootkitRevealer came out, the malware said, oh, shoot...

Leo: Something's looking for us.

Steve: ...this is RootkitRevealer making the call. Let's tell it the truth. Show it our files.

Leo: Right.

Steve: Because RootkitRevealer doesn't know about specific rootkits. It's looking for a lie. And so the malware stopped lying to RootkitRevealer. I love that. So what they did was, now it makes up a random name for itself and runs as a Windows service.

Leo: So essentially RootkitRevealer is a rootkit.

Steve: It's got rootkit technology, yes, because it's accessing the physical hard drive.

Leo: Right. It has to act like a rootkit to find a rootkit. But we trust Sysinternals.

Steve: Oh, well, those guys are tremendous, yes. Okay. So we're still locked in a cat-and-mouse battle with rootkit technology. The RootkitRevealer, free from Sysinternals, finds 100 percent of the current rootkits that are known to exist. And because it does it by knowing exactly how the FAT and NTFS work, in order to trick it, you would have to literally be able to edit physical sectors on the fly, not creating any breaks, I mean, you...

Leo: Pretty hard to do.

Steve: It's a serious piece of work to do that.

Leo: I mean, somebody may well end up doing it. I wouldn't put it past these guys.

Steve: It's what's going to have to be done if they're going to want to fool the RootkitRevealer.

Leo: Very interesting.

Steve: Now, Microsoft's approach, or any approach - oh, and the guys at Sysinternals understand this. They talk about this is a limitation, and the only way to really do this is being outside the system. So the way this works - and the links we have to the Microsoft Research GhostBuster project, people will be able to read some white papers that give more information about that. The idea is, in the system you enumerate all the

files, all the registry keys and so forth, and basically take a snapshot. Then you boot the winpe from a CD. Basically you boot Windows from a read-only medium. And now you look at the hard drive and, again, look at everything that the hard drive has. You parse the registry hive, you figure out what's going on, and you compare the two. So you take a snapshot that may have lies in it. And then from outside the system, from a CD boot, you look back at the hard drive, while that operating system is not running at all, and do a comparison.

Leo: So similarly the rootkit isn't running because the operating system hasn't booted up. You're running from a clean system.

Steve: Well, yeah, the rootkit exists only on hard disk as some files. And there's no way for it to hide itself because it's sitting there trying to run next time you boot. So Microsoft recognizes that of course this is not easy. I mean, you know, you would have to shut down your system. Also there's all that pesky licensing problems with Windows. You know, I mean, it's an advantage of a Linux or an open source system because people can have as many copies of the operating system as they want to. Here, over on the Windows side, you know, you have a problem of licensing a bootable copy of Windows. But ultimately, the only way to know absolutely for sure, if rootkits ever get smart enough to bypass RootkitRevealer, is by booting a static system and then comparing sort of an in-system and an out-system snapshot in order to find the differences.

Leo: Is Microsoft offering a product like that yet, or is...

Steve: No. Nothing...

Leo: It's internal right now.

Steve: You know, they know at the research level that's what they have to do. Now, there are some problems...

Leo: Because I could see - and by the way, we have a lot of IT administrators who listen to this podcast because they of course have to fight this fight for real every single day. And I could see an IT...

Steve: Ooh, speaking of which...

Leo: Yeah.

Steve: I'm sorry to interrupt, but I wanted to make sure the IT guys know that Sysinternals allows the RootkitRevealer to be run on remote systems, using that PsExec utility they have. As long as you have access to the system, you're able to launch it remotely. And it's a...

Leo: So you can scan your whole network.

Steve: Yes.

Leo: Yeah. And eventually it would be worth buying an extra copy of Windows on a CD for doing this, if you had a large installation. It's the individual user that's going to be a little hard pressed...

Steve: Right.

Leo: ...to figure out a way to do this. So rootkits are out there. They're well known. The technology that

they use is well known. There's not much we can do to fight that. But there are at least tools that we can use to try to discover them.

Steve: There are two...

Leo: Can you prevent infection?

Steve: Well, there we come to...

Leo: The same way we prevent other spyware and viruses, yeah.

Steve: Yeah, there we come to the issue of hardening the OS. And right now we have extremely soft operating systems. I mean, the traditional wisdom is, don't run as administrator because a non-admin account, you know, like the guest account or a lower privileged account under Windows, or even, you know, any of the open source OSes, presumably you have lesser privileges. The problem is all kinds of things break. And much software requires admin privileges in order to install it.

Leo: Mostly on Windows, and that's because of the heredity - or the legacy, I should say - of Windows 3.1 and 95 and 98.

Steve: Well, and for example, RootkitRevealer needs admin access.

Leo: Right. Well, that's appropriate. I mean, if you're going to install software, you should be an admin.

Steve: Yes.

Leo: If you're going to run RootkitRevealer, you should be the admin.

Steve: Yeah. What I see Microsoft doing - and I think Longhorn, now called Vista, has some of this. They're working to make it more feasible to run the system as a non-privileged user. So you're able to get elevated privileges briefly in order to run an install of some software, but you never yourself have to log on as that privileged user.

Leo: Every UNIX book I've ever read, every manual, every program, every UNIX sysadmin has always said, "Never run as root." This is one of many reasons why you never want to run as root. You never see that in Microsoft's documentation. Don't run as administrator.

Steve: It comes. I mean, it comes with full privilege. Who would ever, you know, Mom and Dad would never know that they have to lower their privilege because Microsoft knows too many things break.

Leo: You can do that. I set up my Mom on Windows 2000 as a limited user. And, yeah, she would call me, saying I want to...

Steve: Leo, Leo...

Leo: ...you can't update her antivirus. You can't update your own antivirus. And it seems like that should be - at least you should be able to run that. But anyway, but yeah, but that's better than - and

she never got any spyware and never got a virus because she was never running with high enough privileges. You can't install software, you can't install a virus.

Steve: And Mom probably needs to surf the Web and to check email. And so you really want her to have a safe system. And so she, you know, she - not being a power user, she's not going to hit that barrier of needing admin.

Leo: And at home my family all runs their own Macs, but they all run as limited users. It's a little easier on OSX to run as a limited user long term.

Steve: Right.

Leo: It's really hard on Windows, you know, unfortunately. Well, anything else to say about rootkits? Have we covered the topic? Just run to Sysinternals right now.

Steve: Go to Sysinternals, RootkitRevealer. Also, just for the sake of completeness, people will see these links on both of our pages. F-Secure.com also has in beta now something called Blacklight, which is the same kind of thing. It is a rootkit-revealing, rootkit-finding tool. So, you know, the good news is, you know, again, it's a cat-and-mouse game. The spyware has upped the ante, elevated the stakes, is using more technology to hide. So we're beginning to see, you know, the whitehat guys are responding. Unfortunately, we're reacting instead of being proactive.

Leo: The only question I have is, if you detect a rootkit, how do you remove it?

Steve: That'll be a topic for another Security Now!. I mean, yes. I would say, if you detect a rootkit, you'll have to figure out what it is. Then you can Google and find instructions for...

Leo: But if you can't - if no normal tool can see it, it's going to be darn hard to remove it.

Steve: Well, that's exactly the problem. So that's where you really do need to be able to have, like, a multiboot configuration where, again, you boot off a CD, yeah, or take your drive and make it the D: drive of another machine. Then it won't be running, and you can go and delete those files.

Leo: [Indiscernible] won't do it.

Steve: No. No, because it's still the OS.

Leo: Steve Gibson, you've terrified us all, but I think it was an important thing to talk about.

Steve: Yeah, that RootkitRevealer, we'll be showing it on Call for Help, in fact. It's just a very cool tool.

Leo: Let's see how many of our machines that call for help are compromised. What do you say?

Steve: Ooh, that'd be really neat.

Leo: Oh, you think? Thank you, Steve Gibson, and thanks, of course, to our friends at AOL Radio for providing us with the bandwidth for this podcast: AOLmusic.com. And of course you can tune in their

podcast channel on AOL Radio and hear Security Now!, along with some other fine podcasts. We'll see you next time. Have a good trip back.

Steve: Yup.

Leo: Thanks.

Steve: Thanks, Leo.