SECURITY NOW!

Transcript of Episode #176

"DropMyRights"

Description: Steve and Leo delve into the inner workings of a free, easy to use and useful yet unknown Microsoft utility known as "DropMyRights." It can be used to easily run selected, dangerous Internet-facing applications - such as your web browser and email client - under reduced, safer non-administrative privileges while everything else in the system runs unhampered.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-176.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-176-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 176 for Christmas Day 2008: DropMyRights. This show is brought to you by listeners like you and your contributions. We couldn't do it without you. Thanks so much.

It's time for Security Now!. Yes, the show that never sleeps. Yes, a Christmas Day Edition. I'm Leo Laporte, and Steve Gibson is with us. And just in case you're worried, we did not record this on Christmas Day. Hi, Steve Gibson.

Steve Gibson: Hi, Leo. Great to be back with you.

Leo: We are not working on Christmas. We're taking some time off. You're going to be with your family for Christmas?

Steve: Yes, I am.

Leo: Good. Oh, that's nice. Me, too. I'll be in France.

Steve: Yes, you'll really be with your family.

Leo: Yeah. We miss our daughter.

Steve: Reunited for the first time in a while.

Leo: Oh, let me tell you. We really miss her. It's been hard to have her in France for a whole year when she's only 16. We didn't expect this. But she's having a great time. And I'm sure we'll come back reporting wonderful things. And with lots of pictures. I'm bringing the new Canon 5D Mark II with me, so I'm going to have some beautiful shots of Paris at Christmastime.

Steve: Very cool.

Leo: So what possible security topics could we talk about today?

Steve: Well, we're going to discuss some technology which was rather quietly added to Windows XP when XP was released. And a Microsoft engineer who is very security aware made a blog posting about four years ago where he talked about a little utility he had created called DropMyRights. And it takes advantage of some of the new technology. Remember that Microsoft was ballyhooing all the great security features in XP, and how it would be by far the most secure operating system they had ever made. Of course that ended up not being true. And I argued at the time that that's not something anyone can say ahead of time. It's something that only history can demonstrate. And in this case I was correct. XP initially was a catastrophe.

But there is a very cool technology. And this is sort of a - I guess I would call it a poor man's - a very poor man's - Sandboxie. But it's a technology that allows individual applications to not run with admin rights, even though you are running with admin rights. So I'm going to explain what the technology is in XP and talk about what DropMyRights does, how it works, how to use it, for - certainly for the people who don't have the opportunity to use Sandboxie or for people who are really belt-and-suspenders people. I'm using DropMyRights and Sandboxie because - at the moment, I should say - because it's even better than one or the other by themselves. But I've also had some dialogue with Ronen, asking him why exactly doesn't Sandboxie have this built in? And so we're going to talk about that.

Leo: Great, I can't wait. That sounds really, really interesting. All right, Steve. Now, we are prerecording this several weeks before the show airs. So any security news we give you is going to be, well, not news.

Steve: Although there were some interesting things that I found that I wanted to share with people, just sort of of a security news-y nature, without being, like, oh my god, you've got to immediately update your firmware to the latest edition.

Leo: It's not breaking news. Yeah, yeah.

Steve: There's been a trojan around for a long time called DNSChanger. And this has been something which likes to get into people's machines and change their DNS settings because - and in fact I've run across some individuals whose DNS settings were changed away from their ISP's default, to some bizarre bad DNS server somewhere. Now, we know from having talked about the Kaminsky problem that getting spoofed DNS is a really bad thing because your browser will show you the proper URL. You'll look at www.paypal.com and see that it's typed in correctly with no spelling typos, and you'll think you're there. But in fact you're not. Your browser is hooked up to a different IP for PayPal.com because it got the address from a spoofed DNS, either because you're using your real server that is carrying spoofed information, and we know that today only - well, only. About 75 percent of servers have been patched to render spoofing much more difficult. But that leaves 25 percent on the Internet that are still able to be spoofed.

Leo: That's just stunning that that number is still so high after all this time.

Steve: Yeah, it is amazing. And I'm glad I've invested as I have in my DNS spoofability test that we'll be showing to our listeners here in a couple weeks.

Leo: Yeah.

Steve: But the other way this could happen is if your system has been reconfigured to go off to a deliberately malicious DNS server. You wouldn't know it. And in fact that DNS server could serve valid DNS except for those specific domains that it wishes to commandeer. So, for example, if you gave it MyHappyPuppy.com, it would send you back the IP, the proper IP of MyHappyPuppy.com. But if you give it BankofAmerica.com, it sends you to its clone of the Bank America site and collects your credentials and performs all kinds of bad stuff. So this is a problem.

Well, what's new about this DNSChanger trojan is fascinating. The latest version is installing a valid NDIS driver. NDIS is one of the lower networking layers that's always been part of Windows networking. It's using - what I read said that it was installing a driver that had been used by ARPNET, so it's a valid driver. What the installation of this driver does is essentially gives it raw socket capability. Well, what it's doing with its raw socket capability is pretending to be a DHCP server. So one machine on the LAN gets infected with this trojan, which installs this networking driver, which then listens for any other system on the LAN to come online. The first thing the system does as it's booting is it sends out a broadcast, a DHCP query essentially, saying hey, I don't have any IP settings. I need them.

Well, what happens is the trojan hears it, as does the real DHCP server. Now there's a race. Who can respond more quickly? And unfortunately many DHCP servers are running on underpowered machines because they don't need much power. If there's a machine that is fast, that's got this trojan in it, it will respond first, satisfying the outstanding query and essentially reconfiguring the IP settings of any DHCP client to route all DNS to a malicious DNS server. And this affects any LAN environment. So Wendy could set up her laptop at Starbucks that has this trojan on it and be sitting there happily computing, not knowing that this is present. Anybody else who then boots their system at the Starbucks network, because they're essentially on a LAN even though it's WiFi, will get themselves redirected from the actual LAN's DNS server to the malicious one. And this has been found in the wild, and it is going on now.

Leo: Now, we should emphasize, it's not automatic if you're on a Starbucks server.

Steve: No no no no no. Nothing against Starbucks. I'm drinking their coffee every morning.

Leo: We don't want to start any rumors here.

Steve: No.

Leo: But if you've got this bug on your system is the point.

Steve: Yes. So the idea, though, I mean, I use that as an example to indicate, for example, that even roaming users, anybody, any system that is using DHCP, as is the default configuration for all Windows systems. You know, in my own network I've got static IPs for everything, so I'm manually configuring them, and I keep a whole log of every IP that I've configured. But default Windows configurations just use DHCP. That's obtain IP address automatically setting, which is the default. And it means that, if there were a rogue DHCP server anywhere on the LAN, it would receive the request, which is a broadcast out on Ethernet. There are a set of MAC addresses reserved for, you know, that everybody will receive. So not only does the DHCP server receive it, but anybody else on the LAN receives it. Now, typically it's just ignored. But if you've got this trojan on your system, thanks to raw sockets, it's able to intercept it and to build - essentially it's spoofing the reply from the real server and providing the DHCP information to the requesting client. And in this case it's malicious. So you don't want to let one of these things get anywhere in your network because it's not just then affecting the one machine, it's affecting all of the machines.

Leo: All right. Hey, do you recommend, do you think - uh-oh. I'm getting a little error message right over your face. That's not very attractive. Do you recommend a static IP over DHCP in general? I mean, it would avoid this problem.

Steve: It would avoid this problem. I guess the function of...

Leo: Or would it? Would it avoid the DNS issue?

Steve: Oh, yeah, yeah. Because, yeah, you're not using a DHCP protocol at all.

Leo: So nothing is setting your DNS; right, yeah, okay.

Steve: Right. You're establishing it manually. I mean, I know people that are into their network like to know what the IP is of their different machines. I've got three TiVos, and I'm able to connect to them remotely. I have got web browsers, or web servers running in my TiVos, and also some other third-party hackery that allows me to go in and delete stuff that accumulates, and play games. So I know what those IPs are. I know the IPs of pretty much all the many machines that I've got on my LAN. So for me it's very useful to have them fixed.

And there are lots of instances where you do want a fixed IP for a machine. The way you would normally do that, still using DHCP, is you are able, for example, in any of the contemporary consumer routers, to map a MAC address - that is, the physical address of the adapter - you're able to map a MAC address to a fixed IP so that a computer, a given computer, when it asks for DHCP - it asks using DHCP for its IP, its local IP on the LAN, the router will say, oh, this is a MAC address I know about, so I'm always supposed to give it the following IP. So you are able to, like, use DHCP, which is a dynamic IP assignment, in a static fashion so that given machines always receive the same IPs. So you can sort of achieve the same effect. That's often used, for example, when you're doing port forwarding. You want to forward incoming data to a specific machine behind your network. Even though it technically has a dynamic IP, you've told the router always give this particular machine, recognizing it by its MAC address, the same IP.

Leo: Reserve this IP for this machine.

Steve: Reserve the IP.

Leo: Yeah, because we have this debate here. I don't want to have to put - if I put static IPs in all of the machines on our network here, which there are maybe a dozen, and then another machine comes in, I don't want to have to keep track of all that. So DHCP is so much easier.

Steve: Right. It requires a lot of juggling. You have to have, like, a reason for doing it, or just be an old fart like myself. It's just nice to know...

Leo: Or here's a reason. We just gave you a reason.

Steve: Yeah. In some interesting security news, I noted that Firefox, the final version of Firefox Version 2, is scheduled. It's six days in the future from when we're recording this on December 10th. It's scheduled for December 16th release. That'll be the final Version 2 of Firefox. And it loses its antiphishing features.

Leo: What?

Steve: It's losing it because it's still using the very first version of the antiphishing API, which Google has told the Firefox folks they're no longer going to support.

Leo: Well, that's a step backwards.

Steve: So, well, but Firefox Version 2 is a step backwards itself.

Leo: You shouldn't be using it, yeah, yeah.

Steve: Version 3 has been out now since June. I've switched. Everyone I know has switched. I would advise, if you have the ability to switch, you probably should. If you are depending upon the antiphishing feature, which is certainly a nice feature of Firefox, you may want to deliberately not update to the very latest Version that misses that feature. Although you will be informed, if you attempt to update to this final one, that if you do so, if for whatever reason you choose not to go to Firefox Version 3, you want to stay with 2, that you'll be sacrificing their antiphishing interception if you switch up to the very final version of Firefox 2. And remember that then you're on an unsupported - from that point on you're on an unsupported version that will no longer be receiving any security updates or other updates. And so if nothing else has talked you into moving from 2 to 3, then I would imagine that would because, you know, you want to be able to be using a continuously supported browser. And 3 is working just fine for everybody.

Leo: Oh, I love 3. 3, in my opinion, is an improvement.

Steve: Yeah.

Leo: Over 2.

Steve: And I've got a little YubiKey news.

Leo: Okay.

Steve: YubiKey is now being used to authenticate with TrueCrypt. TrueCrypt of course we've talked about is the terrific utility for encrypting deeply and very strongly encrypting drives or whole drives or partitions and directories and so forth. It's a very flexible tool. There is a personalization tool that Yubico, the creators of the YubiKey that we've talked about several times, have now produced. It was originally just an ActiveX control. Now they've turned it into a little turnkey Windows app. It allows you to do two things. It allows you to basically take over responsibility for your own YubiKey and give it its own new AES key, which is what it uses for generating the one-time only tokens. So that allows you essentially to take control away from Yubico's authentication servers, if for whatever reason you want to do that for yourself. The other option, which is interesting, and this is what TrueCrypt is using, is it allows you to create a 32-character static password. That is, change the way the YubiKey works entirely so that, when you press the button on the little USB YubiKey dongle, it spits out a long and absolutely random, but never-changing password, 32 characters long.

And what that's useful for is any instance where you need to do offline authentication. For example, if you wanted to use whole drive encryption, like on a laptop, so that nobody can use your laptop unless it's you, well, you could use TrueCrypt in order to produce the whole drive encryption, which is doing preboot decryption and preboot authentication. The problem is there's no way at that point to have Internet access, so you couldn't use the normal YubiKey. So you could use instead this YubiKey which you have changed using their personalization tool to have it emit a 32-character monster random password. People might say, wait, wait a minute, only 32 characters. Is that enough? Well, yes. I mean, if it's ASCII...

Leo: It's pretty good if it's really random.

Steve: Yes. Exactly. It'd be absolutely random. Imagine that you probably get about 64 bits, that is, 64 different characters per - 64 different characters per character slot, so six bits. So that's going to be times 32, which is what? Is that 192, I think? So that's 192 random bits. I mean, that is seriously good protection, even though it's not changing. Now, the vulnerability is it's a one-time - it's the same thing every time. On the other hand, even if somebody saw it, and any time you're typing in a password, it's blanked. But even if they saw it their eyes would just glaze over. But passwords are always entered blanked out, so no one is seeing it anyway. So it's just an interesting...

Leo: How hard is it to change? I mean, if you decide that you want to go to a different...

Steve: You can change it any time you want to. The new personalization tool allows you to...

Leo: I love this.

Steve: Yes. I think it's very cool, Leo.

Leo: Because you could use it for your WPA key...

Steve: Yes, exactly. You use it for WPA. The only sensitivity would be to beware of keystroke loggers. A keystroke logger is not a problem with preboot authentication because nothing's running at that point. You know, there's no OS or anything going at that point. It's just TrueCrypt. But you're right, Leo, I mean, you could use it to instantly load your gnarly WPA key into a system that you're visiting. Or say, for example, some friends come over, I mean, how many times...

Leo: I can have a YubiKey and say here's the key.

Steve: Yes.

Leo: Just put this in.

Steve: Yes.

Leo: I don't have to divulge it. They won't see it.

Steve: Yes.

Leo: That's what I'm going to use my YubiKey for.

Steve: It's a perfect solution because we've always - many users have said, hey, I've got a gnarly password, but how can I give it to my friends in a safe way? So they bring their laptop over. You simply put the cursor in the field where it's asking for their password. You touch the little button on the YubiKey, it zaps it in. Then it says confirm it. You do it again, zaps it in, they're now on your network. And they don't know what it is, and it was never disclosed.

Leo: I like that.

Steve: It's really neat.

Leo: They're going to sell a lot of YubiKeys with stuff like this. This becomes really great.

Steve: Yeah, I mean, and again, remember that the YubiKey is not expensive. So anyone can get it, use the personalization tools to turn it from a one-time password into this 32-character random password that you can use anywhere you want something that makes your eyes glaze over when you see it. And no one is going to be able to glance at it and then type it in.

Leo: That's just awesome.

Steve: And, finally, the open source Password Safe utility now supports the YubiKey in its full, normal, using Yubico for authentication, one-time password mode. There's been some dialogue back and forth. One of our listeners who is a YubiKey user wrote to the author, the maintainer of Password Safe, and said hey, is there any way you would support the YubiKey? And they now do. So the Password Safe, which is, I think - I'm not sure if it's multiplatform in the YubiKey support because he's got two versions in beta right now. One adds the YubiKey; the other is Linux, that was doing some experimental stuff. I don't know if the Linux one incorporates the YubiKey support also. But YubiKey support is now part of Password Safe. So that's very cool.

Leo: That's even better because now, okay, I use that strong password on the YubiKey to unlock Password Safe. And that could have the WPA key in it. It could also have my True- oh, no, because the TrueCrypt key you need on preboot. So I need three. I need three YubiKeys.

Steve: Yes.

Leo: They're going to sell a lot of them. That's great.

Steve: Yeah.

Leo: That's really great.

Steve: It's a really nice forward motion.

Leo: So you put the Password Safe on your U3, and then it'll say okay, what's the password? Then you put the YubiKey in the other USB slot. It authenticates. Pull the YubiKey, put that back in your pocket, and you're done.

Steve: Yeah. And the way Password Safe works, as you may know, is that when you minimize it, if you maximize it, it makes you enter it again. And if you've got a gnarly password, that's a real pain. So you can simply use the YubiKey in order to give it the password again. Just zap it right back in again.

Leo: Love it. Love it.

Steve: So it's nice authentication.

Leo: All right. We're going to come back in just a second. We're going to talk about DropMyRights, a good thing. Usually you'd say I want more rights. Not in this case. Fewer rights are better for security. We'll explain all and talk about this. It's free; right?

Steve: And I do have a fun SpinRite story, too.

Leo: Well, do the SpinRite story then. And then we'll talk about DropMyRights in a bit.

Steve: Well, this is a neat guy, Earl Pearce, whose subject in the email he sent to me by way of GRC Sales account just says "SpinRite is amazing." And he said, "Hi, Steve. I purchased SpinRite in 2006 after hearing about it on the Security Now! podcast and used it a few times, just in case, on my system. But no miracles were apparent. However, a few days ago my laptop suffered from a severe case of BSD..." - I think he means BSOD, Blue Screen of Death.

Leo: BSOD, yes.

Steve: "...so bad that I had to video the screen just to read the BSOD unmountable boot volume message." So it must have just been flickering up very briefly for some bizarre reason, and he had to videotape it and then, like, single-frame his video in order to figure out what the message was. He says, "I tried all the usual Safe Mode yada yada tricks with no luck. Then I remembered SpinRite. After a few warnings about drive heat, SpinRite rebooted, and the system worked great. You are a wizard. Carrier of the wand. Gandalf reborn. Creator of the best hard drive resurrection tool in the galaxy. I have listened to the other emails you have received and thought, wow, it must be good. But experiencing the miracle in person was amazing. Thanks for all your hard work and long hours." Signed, Earl Pearce.

Leo: Wow. It must be good.

Steve: That was a neat message, and I just wanted to share it.

Leo: Oh, that is. That's a great message. So I didn't realize that DropMyRights was just kind of a side project by a Microsoft employee. I had no idea.

Steve: Well, yeah. I mean, what I like about it, being Mr. Write Everything in Assembly Language and Have Little Tiny Applications, what I really like about it is that it is also extremely small. Now, okay. There's a bunch of links that users are going to probably have to come to Security Now!'s page to find, our show notes for this Episode 176, because the original blog posting from four years ago unfortunately is obsolete. I had to dig around for quite a while to track it down to find where it had gone. Microsoft had moved it over the course of four years. The links that I could find were broken. So I found everything. So there's a bunch of resources relating to this in this Episode 176's show notes. So just go to GRC.com/securitynow and click the third icon, which is the show notes, and that'll take you to the page. Or I'm sure that your team will have also added them to your site, Leo.

Leo: Yes, of course.

Steve: So my point was that it's a very small little utility that basically leverages technology that has been built into Windows since XP. Windows has - and we've never really talked about the security model, the way security functions in Windows. But frankly, every time I go back in there, because I've had to understand it and dig around in there from time to time over the years, every time I refresh my understanding of everything going on in Windows from a security standpoint, I am surprised again that it even manages to boot. There is just - there is so much stuff happening under the covers, behind the scenes in Windows at an amazingly granular level.

There are fundamentally two different things, objects, in the Windows security model. There's this notion of an access token which is a collection of rights. And then there's a security descriptor, which is sort of a collection of needs. And so when you log in, when a user logs in, based on the type of account they have - we had talked about, of course, normal users versus admin users. There's guest users. There's limited users, various types of users. Well, when the user logs in, the rights that they have are collected together. Users can be members of groups. And so groups can confer rights onto users that are their members. So there's a complex set of sort of rights aggregation which occurs when a user logs in.

Now, once they're logged in, any applications, any processes which they run, sort of like from their own context, from the context of their user, they confer all of their rights onto that process. Then anything that the process attempts to do, and I mean pretty much anything, it's checked against sort of like the rights requirements of the things that the process is sort of doing it to. So, for example, files have the so-called "security descriptors." Directories do. Other processes do. Individual threads of execution within the process, individual execution threads have the security descriptors. As do registry keys and Windows services, and even printers have them, and network shares, and interprocess synchronization objects. Basically sort of everything that is something in Windows has its individual set of requirements for who gets to mess with it - who gets to read it, who gets to write it, who gets to change it, who gets to even browse it, even see that it exists. And so there's this high level of granularity. And amazingly, when a process is rummaging around doing things at very high speed in Windows, somehow Windows has time to check all of this. And, I mean, on an individual action-by-action level to make sure that the user that started the process that's trying to do whatever it is has the right to do so.

Now, there are a huge, like, dictionary of sort of privileges that can get conferred on a process. For example, like there's a restore privilege and a backup privilege, a load driver privilege, a shutdown privilege, a debug privilege. Even a privilege to change the system time so that if - so that you could create a user who could do everything except change the time on the system. Just when they try to do that, they can't. They're unable to do that. A debug privilege is very powerful because debugging allows a process to view into the memory of another process, which is what a debugger that is running in Windows inherently has to do, is be able to peer into and even go in and muck around in another process's memory space.

Now, backing up and restoring might sound sort of benign. But what the backup and the restore privileges confer on a process running with, like, backup and restore rights, is they violate any other access restrictions for that program. Meaning that, for example, if you want to back up a whole system, well, the whole system might have a bunch of system files and a bunch of different user, multiple different users' files. And when you back that hard drive up, you want to know you're backing it all up, you're getting it all. Which means that any other sort of restrictions which would prevent one user from being able to see or view another user's files, well, the backup process needs to be able to see everything. So the backup privilege is very powerful.

And of course there's sort of the granddaddy of all is the admin privilege, the overall administrative rights. Malware wants to have admin rights because, for example, admins are able to create files in the system32 directory, which is obviously a highly privileged directory. You don't want malware to create files in your system32 directory. Malware is able to - I mean admins, somebody with admin rights can terminate a process, meaning that if something were running that an admin had created, it would be able to terminate whatever process in the system it wanted to. Admins can also disable the Windows firewall. Non-admins are not able to do that. Well, you don't want malware to be able to turn off your Windows firewall if it wants to. And even downloading and writing files to the system32 directory. Or, for example, modifying or deleting keys in the registry. There's a whole branch in the registry called HKEY_LOCAL_MACHINE. Admins have access to that, but non-admins do not. And again, you don't want malware to be able to go in and delete keys from the registry because those are, for example, where things you care about are used for configuring and starting them up when you boot Windows.

So there's this notion of rights that the user has, and then the things that require the processes that are running on behalf of the user to have in order for the operation to be successful and to not generate an error back through the operating system to the application. So what Windows XP added was the ability to, on the fly, allow these rights to be modified. There's something, a new feature in Windows XP which is called Software Restriction Policies. Software Restriction Policies essentially allow policies to be applied based on, like, the hash of an executable, the executable's name, the location of an executable on the hard drive. So you're able to sort of create broad-based policies which corporate IT professionals will use in order to sort of lock down the systems and circumscribe what their users can do.

Well, this security guy at Microsoft, Michael Howard was his name, who did this blog posting four years ago, he said, you know, I'm an admin. That is, I want to run with admin rights because of the stuff I'm doing. I'm a power user. I don't want to have to be logging in and changing - logging out, logging in, changing who I am all the time. And we know historically it can be annoying to be running as a non-admin user because of sometimes there are things you want to be able to do like, for example, installing software.

Leo: I would guess most of our users are the type that just say, I want to be the admin. I'm not willing to be a limited user.

Steve: Yes, exactly. And so what he realized was - that is, Michael Howard realized that he could leverage this rich security model in Windows and the new feature of software restriction policies to create a simple little app that would allow some programs to be demoted from admin rights, even though he was an admin. So normally, as I was explaining, if you're logged in as an administrator, anything you run, anything that runs from you is essentially - it inherits the rights that you have. But he recognized that some things were extra dangerous. And sort of this begins to overlap on Sandboxie's territory because he realized, for example, IE, you really don't want Internet Explorer to run as an admin because then anything that it runs has full administrative privileges. And as I was saying, can write to the system32 directory, can terminate processes, can alter the settings to your firewall and so on. So you'd really want to be able to run some applications with limited rights.

And thus was this concept he came up with of DropMyRights. The idea is, it's a simple little utility where you simply - if you had a shortcut on your desktop, for example, or down in your Quick Start tray, or under your start menu, if you right-click on the shortcut and look at the command it's executing, you'll see some string. It'll say C:\program files\whatever it is, the path to the executable that ends up with iexplore.exe or outlook.exe or firefox.exe or opera.exe or outlook.exe. Anyway, you're able to prefix that string with dropmyrights.exe. So DropMyRights is actually the thing you're running, and you're passing it the path to the executable as its argument. So what happens is it runs, and it creates a process token that has non-admin rights. That is, essentially, normally a process token created, for example, the one that it's running under, would have your rights. It would have your full administrative rights. DropMyRights creates a token that has non-admin rights. And then it runs the thing that you would have normally run under those restricted rights.

Leo: So it's interesting because this is kind of like the Run As command in reverse.

Steve: Yes. That's a very good analogy. It's like Run As. Whereas instead of, like, being a limited user who wants to run as admin for something, you're saying I'm normally an admin. And the beauty is you're able to set this up with your shortcuts so that any time you run Firefox or Outlook or Opera or IE or whatever, it's running with reduced rights.

Leo: I want to talk about this a little bit more, and the implications of it. And I really want to talk to you about the difference between running - what I do, and what most people in Linux and UNIX do, which is run as a limited user, it's kind of the conventional wisdom. You never run as a superuser on Linux. And we use this escalation, this Run As, or in the case of Linux SUDO, or sudo, to escalate, versus this de-escalation process, which is probably pretty typical Windows user thing. From the day I started using UNIX, and then later Linux, it was drilled into you, you never ever log in as superuser. You always log in as a normal user. If you should ever need administrative privileges, Linux makes it very easy with the sudo command to do that, to escalate. Wouldn't it be better to do it that way on Windows?

Steve: Absolutely would be. The problem is that the two different operating systems, Linux and Windows, sort of came at this from different directions.

Leo: That's a good point because Windows 95/98/ME you were always administrator. There was no conception of a separate limited user.

Steve: Right. And essentially it's Microsoft has been trying to impose the security paradigm after the fact. Whereas UNIX, this notion of being a root user and having godlike capabilities, I mean, there's always been, from day one there's been an appreciation of the need for security and the notion of user accounts and the idea of a superuser who is absolutely not the normal user that you run as. You only go into superuser mode when you're wanting to be an administrator. So there you elevate yourself.

Well, the point is because the system always evolved that way, all the applications running in a UNIX environment made the assumption that their users would not be root, would not be the superuser; that they would be a regular, normal, lesser privileged user. Unfortunately the assumption was reversed under Windows. So applications assumed the way Windows always was, which was the user was the user, there's only one type of user. This notion of creating a lower privileged user was something Windows later in the game said, oh, that's kind of a good idea, let's do that. The problem is that most of the developers and programmers, they were all running as admin. So none of their software was being tested as non-admin. So they were making an implicit assumption that everybody was going to be using Windows the way they were. And it turns out then that trying to use their systems as a non-admin created all kinds of problems.

Leo: Well, I guess it really goes back to DOS. When, in fact, in the earliest days of DOS you didn't even have other - you not only didn't have other users, you didn't have other processes. Before TSRs, there was one thing going on in that computer. And DOS was so brain dead, and that really is the heritage for this whole thing. I mean, it began there. Whereas UNIX always was multi-user. And anytime you have a multi-user system, you have to have concepts of permissions and different levels of power and so forth.

Steve: Well, and so what Windows has, Windows with this use of, this notion of software restriction policy, Windows defines five different levels. There's the trusted user, the normal user, a so-called "constrained" user, an untrusted user, and then disallowed. And so those are the five different levels that an application can run under. Trusted is essentially admin, meaning that the application gets to run with any rights that you otherwise have. Those are conferred on the application. Then the normal user is the so-called "non-admin." And people who use DropMyRights will see that there are two options. In addition to normal, you can ask DropMyRights to drop them to even below those of a constrained user, I mean, below those of a normal user to a constrained user, or to an untrusted user. Well, untrusted is virtually useless. Those rights are so restricted, as I was experimenting with this, not even Notepad will run as an untrusted application.

Leo: That's pathetic. What can you do with Notepad?

Steve: Exactly. So untrusted, I don't think anybody will ever be able to get anything to run as an untrusted user. I was experimenting with constrained. And even that is kind of sketchy.

Leo: Is that because these programs feel like they have to write to, I mean, why couldn't they run? I don't - what are they doing?

Steve: It's because they're not developed with this whole mentality of minimalism.

Leo: They probably write to the registry, for instance, which would be something you wouldn't want an untrusted program to do.

Steve: Well, for example - that's a perfect example. A constrained program, if you use the argument C with DropMyRights, then for example the whole key in the registry underneath HKEY_CURRENT_USER, which is a very rich key, full of registry entries, it's read-only.

Leo: Oh, there you go.

Steve: So no constrained program, that is, program running as a constrained user, is able to modify anything underneath the HKEY current user. And similarly, anything under the so-called User Profile Directory, that's Documents and Settings\Administrator, that's completely inaccessible to a constrained program. And even crypto operations, including SSL negotiation, do not work. So that just tells you you can't run a browser under constrained. So I don't want people to be too concerned about those other two options. Basically just using DropMyRights and dropping the application rights to normal, that takes away admin privileges, which takes away basically all the things you don't want a program to be able to do. And so, for example, I've been using DropMyRights now for the last few weeks [inaudible] discovered it. I've got Eudora running as a non-admin and Firefox running as a non-admin. And they work just fine as a non-admin. Yet the rest of the time I've got godlike rights on my system, which I need because I'm using...

Leo: You're using Notepad.

Steve: I'm using Notepad. Now, here...

Leo: This cracks me up.

Steve: Here's what's really interesting, is when I - again, when I ran back across this, and I think it might have been, like, two weeks ago a Q&A question, because I've seen people talk about DropMyRights, I know that you and I have talked about it a couple times, Leo, but it's not something we've covered extensively. I was looking at this, thinking, you know, why doesn't Sandboxie just add this, because this seems like a good thing for it to do. So I shot Ronen a note. And I said, hey, what about this?

And he said, well, first of all, Sandboxie already strips many rights from the programs in the Sandbox. It strips the restore privilege; the backup privilege, which we talked about as being very powerful; the load driver privilege, so that nothing running from anything derived from a sandbox can load a driver, and that's good because you don't want to be able to do that. The debug privilege, which we talked about, is very powerful. You absolutely don't want a program to be able to reach into other process spaces and muck around with it. And I'm not sure why, probably just because it was there, he removes the system time privilege so that something running in a sandbox is unable to change the time of the system. And I said, okay, cool. What about admin? And he says, well, I really don't want to do what something else has done. And I said, well, I kind of appreciate that. But I think you ought to put it in there. And so it's in the beta right now...

Leo: Oh, that's neat.

Steve: ...of Sandboxie. I don't know whether the beta will be public. I think it's 3.3.3. I'm using 3.3.2. Or maybe it's 3.32. But it's in the next one. He sent me a link to it privately so that I could play with it. But there's now an option that will be in the next version of Sandboxie that incorporates the full strength of DropMyRights, which Ronen has completely figured out and understands, so that you can optionally run anything in a sandbox in a non-administrative context. Now, it's worth, however, talking about DropMyRights for, for example, all of the 64-bit users who are unable to use Sandboxie. So this is still something moving forward for people who, for whatever reason, aren't using Sandboxie, don't want to use Sandboxie. Still makes a huge amount of sense to run. If you're running as an admin normally, by all means look into DropMyRights to run specific dangerous applications, that is, anything that is Internet facing. Your email and web browsing, maybe newsgroup reader, instant messaging program, whatever. If you like to run as an admin, I'm not going to scold you about it. I mean, I do. Then this allows you to bring the rights down of other dangerous programs so that there's less damage that they can do if something tries to take advantage of them. And it's completely free.

Leo: And because Run As is so inconvenient, people are just not going to use - are not going to run as limited users. I have to say, though, it is easier on Vista than it used to be. I mean, it's certainly - it's not doable on XP, but it's not so bad on Vista to run as limited user.

Steve: Well, and Vista sort of incorporates this. When we were talking, the times that we have in the past talked about this notion of privilege elevation, it's when we've talked about User Account Control, the UAC in Vista. Because in Vista when you log in, remember that the user gets two sets of credentials. You get a limited credentials, which is how you normally run. And there is the option of being elevated temporarily to privileged credentials, and that's what the UAC essentially allows you to do is lift your credentials briefly to do something that you normally don't have permission to.

Leo: Right. So that's very similar to this. Well, it's the flip-around, but it's...

Steve: It's sort of the evolution of it. But for those of us who are not on Vista...

Leo: Still using XP...

Steve: Have no plans to go to Vista, this is a tremendous, nice, little, I mean, it's clean and simple, nothing's like - there's no weight to this at all. It simply launches these apps with reduced rights.

Leo: Very cool. Well, it sounds like a must-get. Absolutely free. Does it run on Vista at all? Probably not.

Steve: Don't even know. I don't care.

Leo: You don't need it. You don't really need it.

Steve: I don't care, Leo.

Leo: You don't care. You're happy, and you're going to use it as Sandboxie, anyway, when [indiscernible] comes out [indiscernible].

Steve: Exactly. It's going to be built into Sandboxie here within a few weeks probably. So maybe by the time this episode is being aired. I want to just close by saying something that I alluded to before. I think there's a way to do this with policies and not even needing the DropMyRights.exe. During the research I was doing, I said wait a minute.

Leo: That would make sense.

Steve: Normally there's only two categories, trusted and disallowed. But I think there's a way to create the additional categories. And, if so, then the policy system, which automatically - which, like, corporate IT uses for constraining what their users can run. I think normal users could use this and just automatically have a class of applications get non-admin rights, even if they're running as admin. I'm going to - I don't think I can hold myself back from poking around at this a little bit more and seeing if maybe I can come up with...

Leo: That would make sense. I mean, I think you should be able to. And, you know, you may not have to poke around too much because I bet you we've got some IT types who are running their office exactly that way. I mean, that's what I would do is use GPEDIT and just say, you know, this is how it is.

Steve: Well, but I've looked. And there's only trusted...

Leo: There's no settings.

Steve: ...[indiscernible] allowed.

Leo: Isn't that interesting.

Steve: It's either yes or no. It's a binary option.

Leo: That seems like something that you really should have, to me.

Steve: I would think so.

Leo: It's very odd.

Steve: Well, although you can imagine, in an IT corporate mode, they're not letting their users run as admin. All their users are being run as normal users.

Leo: Well, right. Can you do it the other way around? Can you say escalate this program? If they have to run Notepad as an administrator, can you have it escalate? I think you can - you know what you do, actually I've done this, is you create a Run As icon.

Steve: Oh, sure. Exactly. And then you provide credentials on the fly.

Leo: You have to give them credentials; right.

Steve: Because Run As is going to prompt you for your username and password.

Leo: Exactly.

Steve: Yeah.

Leo: Oh, what a tangled web we weave when first we operate in DOS.

Steve: So again, if our users will go either to the TWiT.tv show notes or my page, the Security Now! page, Episode 176, I will have a page there with links to track this stuff down. And it's just a very nice, cool thing to do. Drop the rights of high-risk programs so that they're not admin, even if you are.

Leo: I agree. Very cool. Very - a slick idea. And I bet you the inspiration for UAC, at least to some degree.

Steve: Yeah. It was certainly the precursor to it.

Leo: Yeah. You can get more information, as always, on Steve's website, GRC.com. That stands for the Gibson Research Corporation. GRC.com. You'll find, of course, this show, and 16KB versions if you're bandwidth impaired, or the full 64K version. You get the transcripts so you can read along. Many people like to read along while Steve talks. You can also get the show notes, as he mentioned. We have stuff there for you, including a link to DropMyRights. And let's not forget, GRC.com is the home of SpinRite, the world's best file and - or I'm sorry, disk recovery and maintenance utility.

Steve: Yay.

Leo: Yay. Just go to GRC.com, click that big SpinRite link. While you're there you'll see a lot of free programs, too, like ShieldsUP!, Wizmo, Shoot The Messenger, DCOMbobulator, all that stuff. Steve does a great job of giving back to the community. GRC.com. Merry Christmas, Steve. I hope you're having a wonderful holiday. And I thank you for your - just it's been such a great three years working with you. We're in our fourth year now.

Steve: Yup.

Leo: And always been a pleasure. And I just - your determination never to miss an episode, even on Christmas Day, blows me away.

Steve: And I guess New Year's Day, too, next week; right?

Leo: Next week, New Year's Day. New Year's Day. Yeah, we'll do it later in the day, give you a little time to recover from New Year's Eve. But we'll have it out, of course, for you in time to enjoy a little revelry on New Year's Day. Steve, have a great holiday. And we'll see you in 2009 for another great year of Security Now!.

Steve: Thanks, Leo.