RunAs Radio

http://www.runasradio.com

RunAs Radio is a weekly Internet Audio Talk Show for IT Professionals working with Microsoft products. The full range of IT topics is covered from a Microsoft-centric viewpoint.

Text Transcript of Show #170
(Transcription services provided by PWOP Productions)

Dana Epp Fixes a Security Vulnerability!
July 28, 2010

[Music]

Brandon Wenn: From runasradio.com, you're listening to RunAs Radio, the Internet audio talk show for IT professionals with Richard Campbell and Greg Hughes. This is Brandon Wenn, announcing show #170, with guest Dana Epp, recorded Thursday, July 22, 2010. RunAs Radio is produced each week by PWOP Productions, providing professional media and podcasting services online at pwop.com. You can follow the boys on Twitter at twitter.com/runasradio.

Richard Campbell: Thank you, Brandon. This is Richard Campbell. With me as always, or at least most of the time, my good friend Greg Hughes.

Greg Hughes: How are you doing?

Richard Campbell: I'm well, sir. It's summertime. What am I doing? I'm smoking ribs, that's what I do.

Greg Hughes: Well, good.

Richard Campbell: It's a party every week in the fancy new backyard. I tore apart one of my water-cooled PCs, full of aluminum oxide in the water lines. Not good, little bits of white debris. Yeah, I'm not happy about that. I'm going to have to disassemble the water jackets to clean them all out. It's bad.

Greg Hughes: Very cool. See, my version of playing around with the water, and summertime is get out and just use your boat or some jet skis and go out and avoid computers like the plague.

Richard Campbell: Yeah, there you go. It's just a good time to be repairing stuff and I've got to get things fixed up. I'm not traveling so much right now so I'm trying to get gear in good shape absolutely. But hey, let's jump on to the guest because I think we have a really cool story today because I've brought back Dana Epp. So Dana, I don't even know if I want to do this whole bio. You're still the CEO of Scorpion Software?

Dana Epp: I am, I am.

Richard Campbell: So you haven't really change jobs a whole lot. Security seems to be your focus. Got a bunch of cool products out there and before the show we're having a little discussion about the challenges in the development environment. But we had a conversation, you and I, a week ago or so. You're in the middle of a big security panic that I thought would make a very interesting conversation so why don't we go all the way back to the beginning of that and tell me what you've found.

Dana Epp: Okay. So interestingly enough we actually had a vulnerability in one of our products, the part of product that includes some authentication mechanisms that provide two-factor authentication when logging on to Windows Server and workstations, and it ends up that we actually had a customer report to say a problem that they were having which was more indicative of a bigger problem and it ended up being a vulnerability that would allow in certain situations when using the Windows Credential Manager to save CRUDs to bypass the entire plumbing of the API that Microsoft makes available for this stuff that we were using. That became a panic for us because obviously our agents are running on thousands upon thousands of servers and in this particular case it's very isolated. It's all very administrator account on Windows Server 2003 only and only when certain policies are applied. But it's still is a critical vulnerability because used in an inappropriate way it would give you a way to bypass the system which would not be good.

Richard Campbell: So this started out with a customer finding what he thought was a bug in your product.

Dana Epp: Yeah, and ultimately it's a bug in our product. It was a vulnerability that exists because we were exasperating it because of the way we were doing something. But what's interesting is they didn't see it that way. They just thought if this isn't working, can you please tell us what's going on.

Richard Campbell: Right.

Dana Epp: So we leverage Hyper-V very heavily here for everything it do and so when a case like that comes in through customer service, they open up basically a technical research case which then -- what they do is try to repro that and if they can repro that they record that repro in a way so that the developers can see it and then they attach it to an actual developer case and a defect tracking system and at that point we can then replay that at any time because they can share the VHDs and the developers can see what would have caused this and then look at what we need to do to fix it.

Richard Campbell: Right.

Dana Epp: In this scenario they're looking to just do this underlying problem and they're like, you know what? This actually is a bigger problem than the customer realizes and that was when I got involved because at this point it was a security issue and we had to go back to day one and take a look at the threat model and take a look at the entire attack plan of what the issue was and we were able to find it was indicative of a much bigger problem which we had to actually get Microsoft involved in because it didn't just affect us, it actually affects many of our competitors. So interestingly enough we know two of our competitors that had the same problem. So we've released our fix already and they're going to release theirs pretty quick I'm told.

Richard Campbell: So who do you phone at Microsoft when you think there's a security vulnerability in one of their APIs?

Dana Epp: So here's what's really interesting in that scenario. Microsoft has a triage team for security stuff but it's for effectiveness of security stuff over their stuff. So if it's a bug in Office or in Windows itself, you can call them up and they have an instant triage team that takes care of all that kind of stuff. But this is different because this is a third party add-in that belongs to us and we had to go through the traditional developer support channels which actually is quite frustrating for me because being a Microsoft Security MVP I have some internal access to some people which allows me kind of sidestep the standard help cases.

Richard Campbell: Right.

Dana Epp: And I've tried actually using of that, but because of the complexity of what was trying to be accomplished, that didn't go so far. So I went so far as to reach out to some developer evangelists, I reached out to some of the guys down at Redmond. It came to a point where there are only two people currently in Microsoft that are dealing with support of this particular issue because it's Windows Server 2003 legacy stuff.

Richard Campbell: Right.

Dana Epp: So we had to actually open up a proper support case having to go through the standard tier one triage. The good thing was though is I had the full documentation of exactly what was going on and I actually said here's the escalation engineer we need to get to and the nice thing was within 24 hours after that it was to that person and then I had already sent an email in advance saying this is coming to you, as soon as you get this case number please pick it up and he did and we then were able to get right down in there and he was able to look at the code and that was one of the critical component because we knew that what we had found was against what the public documentation was on MSDN for it.

Richard Campbell: Yeah, these documentations are supposed to work this way and it's actually working this way.

Dana Epp: Right. What end up happening was it wasn't a failure in the documentation, it was a lack of documentation because we were -- it ultimately comes down to this one thing. You can set up policy in Active Directory that says always prompt the password. Well, the theory is that it states there that will take care of any remote connections. Well, that's not entirely true. What it is is that it takes all for standard terminal services RDP connections, not console-based remote sessions and that's actually a separate policy setting which is an entirely different area in Windows and so we didn't know what we didn't know and so ultimately what ended up happening was this was an attack that we just simply did not have in our thread model. What's funny is in our original thread model we had explored what would happen in this scenario and what we ended doing was contact Microsoft and asking, "Okay, how do we deal with this," and they said, "Well, you just adjust this policy and this policy will take care of it." I went like, "Oh, perfect. Okay, we'll just do that as part of install and our maintenance and that will take care of it." Well, not quite. We didn't ask the question large enough. We should have asked and where else could that policy be applied.

Richard Campbell: Right.

Dana Epp: And that was what we missed. The fix is actually really more interesting.

Greg Hughes: I noticed one thing. You know, you actually have a thread model which I mean if you're doing authentication and security type software, then one would certainly hope that you would, but it's amazing even in a new software being developed today out there across the industry how few organizations are doing threat modeling.

Dana Epp: Yeah. You know, I get the luxury of obviously working a lot with and learning a lot from Microsoft teams and over those last, you know, oh, it's like eight or nine years now, they've really matured in their own development philosophy in things like the STL Process.

Greg Hughes: Right.

Dana Epp: The security involving lifecycle in Microsoft using its own products have been indicative of -- I'm going to say they've been incubating lots of other technologies. So I've had the luxury of being an early adopter of their thread modeling toolset. They actually have this toolset that makes this look extremely easy. You basically build data for diagrams in what would seem like a Visio-based toolset and then it will help you to generate a list of threats based off of how the data flows in that system and that helps us to explore before we write a single line of code what kind of risks aren't going to exist there and more importantly what can we do to mitigate them. That's important. It's not just security software that's doing this. All software should be doing this, but in our case and as we've seen, thread modeling is not a panacea. It will not solve everything. It will only solve and point out the things that you're very clear about and understanding. It's what you don't know, you don't know and that always bites you which is what got us on this vulnerability.

Greg Hughes: Right.

Richard Campbell: Yeah. I'm just thinking could you have tested it for this? Like it's one of those things where you're trying to find all of the particular combinations, this version of...

Dana Epp: Well, hindsight is 20/20.

Richard Campbell: Oh, absolutely.

Dana Epp: And now we know we could have tested for it.

Richard Campbell: Right, right.

Dana Epp: Because now we know what to look for, but at that time we didn't and that was the thing. That was interesting about the fix. It was because when I -- because I got involved in the fix side of it. I actually went deeper than most people would have to make sure that the class of vulnerability at this gets eliminated. So we actually did API function pointer manipulation in the system so we know what causes the review of Microsoft's team on components to look for this policy, the shoo-ins, and what we did was we basically created a proxy or hijack where we take the function pointer in which Microsoft code is calling into and they think they're calling into the underlying API but they're actually calling into our code, and then we call into the code underneath the hood to get the policy decisions and then we manually adjust it where we have to have it to maintain a secure state and then we passed it down to them. They see no difference. It completely functions properly and all we've done is we've manipulated the function pointers in memory in a way that is stable and reproducible and in a way that we can guarantee that we can manipulate the credentials to struct the way we need to to provide a safe set of policy to the system.

Richard Campbell: And it gets back to also protecting the customers from themselves because they could setup a set of policies there that would undermine your product entirely.

Dana Epp: That's right and it might even be indirectly. Our shop has it where a policy decision in one place may affect the policy elsewhere or worse yet the attachment because that's where we were looking at because as soon as we find something like this, this is where -- I actually enjoy getting involved in this kind of stuff because we start looking at it as the attacker. Okay, so here's all we know. Ultimately, policy decision in Active Directory will boil down to this registry setting. If I have a vulnerability to mount that registry setting remotely, I can change that to this and then what happens? Oh, I can bypass the system. I can bypass any team of a system doing that. Okay, so that's bad. Okay, what can we do to deal with that? Simply changing the policy wasn't enough because it was actually possible for a rogue agent to be able to manipulate that remotely and then get in and bypass. Now granted they still have to have administrative credentials, now we're talking about an administrator to access a point here where we're trying to restrict it. You can't stop it, the administrator, because they can uninstall a product at the end of the day. But we wanted to make sure it was auditable so we know when they do those types of...

Richard Campbell: Right.

Dana Epp: And so by manipulating the way we did, even if they change the policy settings it doesn't matter because we will dynamically, as the policy decision is being made inside the WinLog on process, we actually are manipulating it what we want it to be. So it doesn't matter what it's actually set to, we will always force it to prompt for password. Now, there's a downside to that obviously. If any of our customers is looking to use our product in an environment without a log on, it will not work and that's okay, we document it. If you're using our product, this functionality will not work. And that's a security decision, we're not going to relinquish that decision.

Richard Campbell: Yeah. There's a bunch of functionality that Microsoft has like that too. If you have any of these turned on, auto-logging doesn't work. That's not that unusual.

Dana Epp: Yeah, yeah. Exactly. You know, it's a balancing act. Security is not an absolute. It is about reducing risk to an acceptable position and at the end of the day for what we're doing, and in this case here, it's to provide indemnity insurance when remotely logging in at critical servers, it makes sense. You don't want to have auto-logging capability.

Richard Campbell: Yeah.

Dana Epp: You don't want people to manipulate policy in a way to bypass that, and even if anyone did, you want to know it. So in our case if that policy actually get changed, that gets alerted and bubbled up into the security log and then we have some management packs for a lot of our IT automation systems like System Center and Kasia and LabTec and all this different IT automation systems so that it can automatically alert so that the administrators can know instantly when someone is trying to manipulate policy to try to get around it.

Greg Hughes: Uh-hmm.

Richard Campbell: I'm loving the idea of you had to go through tier one tech support because all of us, all three of us have all been tortured by this.

Dana Epp: Oh, yeah.

Richard Campbell: They do that whole "Thank you for calling (insert name here).

Greg Hughes: The script, yeah.

Richard Campbell: It's like, dude, you really have to get off the script. I wouldn't have phoned you if I didn't need tier two. I can do tier one myself.

Dana Epp: Yeah, exactly. But they don't know that, right? When they call in it's one of those things that they just, "Hey, what's your name? Okay, let me bring it up. Oh, you're a gold partner. Okay, yeah, yeah," and then it's like, well, you know what? Just give me a break. But what's interesting is I'm seeing that Microsoft is seeing it. I don't know if you saw that recent survey that they were asking around but they're taking a look at the whole concept of being able to map your expertise and knowledge-based against wares so you're not wasting time in tier one and it might get you directly to a tier two scenario and that could be useful. In my case it gets worse because Microsoft has a process they have in place. I actually knew who I had to get to but I wasn't allowed to talk to him directly.

Richard Campbell: Right.

Greg Hughes: Right.

Dana Epp: He wouldn't be allowed to take on the case and I understand why because at the end of the day Microsoft is a business and they have to look at the stuff but I didn't care at that point. I did security vulnerability, I needed to address and I wanted to fix it as quickly as possible and from the point of report of the bug to the point that we release a fix is one week and we spent three of those days just going through the Microsoft maze of getting to the right person. Well, let me rephrase that. We were looking for indications from them so during that three days the QA team was testing every other scenario and finding what else they could do so we weren't stopped, we still could keep working but I couldn't release the fix until I knew we had an indicative to what was the underlying problem going on here and we can't look at Microsoft's code but they can so they can say, oh, yeah, okay, here's what it's doing.

Richard Campbell: Right.

Dana Epp: Okay. So then we need to find a better way of fixing it and I'm quite happy with our fix because it removes an entire class of vulnerability here. I don't have to worry about adjusted policy decisions affecting how our stuff works because we will enforce it to a certain way. The only risk we can see to this is if that other agent similar to ours is in there and they try to map the function table differently. But luckily enough Windows Server 2003 has a very static address set for a lot of their addressing so we were able to check for that a little bit.

Richard Campbell: So you haven't actually gotten Microsoft to change any code it has found. You just coded around the problem.

Dana Epp: They did not. Actually what's funny is we've done it in a way so we affect Microsoft's code without them changing a single thing because originally what's going to be, wow, are we going to ever need this issue or hot fix to this, how are we going to accomplish this, but this wasn't Microsoft's problem.

Richard Campbell: Right.

Dana Epp: It was how we were using the system and we didn't know about this policy decisions and in our case what we do is we can do our manipulation and we pass it back to Microsoft's geno which is their components for Winlogon. So what we didn't know was, okay, well, we have an opportunity before the Winlogon processor deals with things to do something. So in our case what we do is I do something, it's to remap the function pointers so that when their code goes to call this particular function, our code gets called and then what we do is we do our math. What's funny is that we couldn't do this in Windows Server 2008.

Richard Campbell: Really.

Dana Epp: It wouldn't be possible because they have defenses to prevent the ability of remapping API calls.

Richard Campbell: Interesting.

Dana Epp: But this particular vulnerability is only on Windows Server 2003 so we're actually using what their protecting game is because it's rogue elements. Imagine if you will that a rogue element remaps API calls to do bad things or keystroke logging or it could be anything. That would be a risk. And so what we did is we used, you know, this is just standard computer software engineering, it's well pointers are pointers if you point them to a different location it will do different things and that's what we did.

Richard Campbell: Yeah.

Greg Hughes: As you've been describing the fix that you did, the one question that comes to my mind then is this a Microsoft supported methodology for doing it?

Dana Epp: Yes, it is. One other thing is when we did this, we made a very -- actually first off, the type of API calls we use to do this is a fully supported API set from Microsoft so this isn't one of those hack it behind the scene kind of thing. We actually use Microsoft API to do this. It's a documented way of giving you a hook into this

particular API calls. So that is completely allowed and more importantly is we got approval from the support escalation engineer that this is an accepted supported scenario and so they just obviously documented for their team that we're doing that and then if there was theoretically someone calling up with a problem even though there's tier three and RH is there, their team knows what's going on and that's okay because that's actually -- it's funny, for the kind of work that we're doing, there are very few companies in the world who's actually doing this kind of stuff this deep.

Richard Campbell: Right.

Dana Epp: So they know who all the players are.

Richard Campbell: Right.

Dana Epp: Just like in AV where they know who all the AV players are. So you know, there are really only about 35 companies worldwide that are doing this specific type of thing. There are hundreds of guys that have kind of build this type of thing but companies doing supported scenarios across thousands upon thousands of servers, they aren't that many. So they know and they've seen this and now it's actually when we found this we actually in our lab we have a lot of our competing products in there that we do for testing especially for things like migrations and stuff like that. One of the things I did was I took a look at two of our competitors and found that they have these bugs though and so we obviously contacted them and we've provided them with the work around because in the interest of our space to make sure that everyone is contacted.

Greg Hughes: Right.

Dana Epp: And they're working on fixes now which is why I can't go into too much detail of what's going on because although our clients are protected they still -- I don't know where they are with their stuff. They're probably a couple of weeks away.

Richard Campbell: But this sounds like it ultimately could end up as a knowledge-based article inside of Microsoft.

Dana Epp: It might be. I know that they're updating the docs to reflect the difference but I don't know if this will become a KB or not. I think the interesting thing is this is such a legacy problem and they're trying to so hard to get people off the Windows Server 2003 in that scenario. The last thing I want to do is spend a lot of time trying to get that out there.

Richard Campbell: Right.

Dana Epp: KB would make some sense on there but again this is a third party property. It's not theirs. It's not like they did anything wrong. It's how we manipulated and use their API and it we had a vulnerability, we had a bug. It was our problem, but what's interesting is we're not the only one. It was an understanding problem in how it works.

Richard Campbell: But it does indicate a potential vulnerability in a Microsoft API.

Dana Epp: Correct. A vulnerable use of the API.

Richard Campbell: Right.

Dana Epp: The reality is and what's interesting is some of our competitors didn't have this problem which means that they have already came across it or they had known, and so that's one of those scenarios where we made a mistake ultimately because we didn't now and for me that was bothersome because we've spent a lot of time architecting these things to make sure it was right.

Richard Campbell: Yeah.

Dana Epp: What was interesting was it was right in what we had information to. And as soon as they made it clear, "Oh, here's what it's doing," we're like "Okay, where the hell that document is. Ooh, okay, all right. So now that's an issue. Okay, fine. We'll deal with that and we'll go forward.

Richard Campbell: Yeah. How were you supposed to know that, you didn't document it.

Dana Epp: And that's common error. There are tons of Microsoft APIs and policies that aren't rather well documented and if you know, you know, and if you don't, you don't. What's interesting is I will bet that we would have never had a real risk to a lot of our clients because this is such a convoluted method of doing things that may not have been a problem but that doesn't make it right because it only takes one person to take advantage of that vulnerability to exploit it.

Richard Campbell: No. And that's what happened. One customer came along and used this convoluted scenario and surfaced the problem.

Dana Epp: Yeah. And so that's the scenario and so at the end of the day we take all vulnerabilities very seriously and we have a policy that really makes it clear that if someone finds a problem, we'll fix it. We just need a chance to do it before you let the world know.

Richard Campbell: Right.

Dana Epp: In this case, the customer didn't realize and when we stopped and explained, you know, "Hey, I appreciate what you've done. I didn't realize what..." I was like, "Yeah, okay and

thanks anyway" and that's the thing. And so it safeguards everybody and that's a good thing.

Greg Hughes: I've got to say I've got to applaud you for notifying your competition of the issue as well so they can put the fix in advance state of the art in their product because that's not something that obviously you have to do but what a great way for you to say.

Dana Epp: Thank you for that. You know, there's not a lot of people who know and other than obviously the listeners here now will know, that's not something we go out and chant and say, "Woohoo, we found something, we went and told our competitors."

Greg Hughes: Sure.

Dana Epp: And this is the belief I have and it's a belief that my team has because I have driven that in our culture. It's that we put the protection and security of our clients before the protection of our profits but still be responsible to our stakeholders, and what that means ultimately is that if we can't solve a problem for a potential customer or we find that we're not able to service their needs, we will walk them to our competitors and the reason for that is that we would rather see them obviously get protected because password security is a very weak form of authentication to start with and we really like to see people using what they feel is the right solution. Secondly, and this is more important than everything, it's that we have -- from a business point of view, we want to see the industry to properly be safeguarded and the last thing we want to see is vulnerability in the types of systems that are being recommended to secure that very infrastructure. So it's in our interest to make sure that they're stuff is working withstanding it all that we, in the place that we're focusing on, we do externally well. We differentiate ourselves in a way that we win in our business because we're good at what we do, but there will be times that people will want to go elsewhere and those might be our very customers and we want to make sure they're protected. So I don't mind letting our competitors know and it actually works both ways. We have competitors bring business to us many times because they deal with enterprises, they don't focus on the SMB space a lot and they don't have support for small business servers or essential business server, or they don't have the ways that tied into direct access the same way we do and that's great and I think that we're lucky to be in an industry like that. Ultimately, I think as a security MVP I also have a higher calling and that's to make sure that we secure the world and it's not just about securing our profits. We still make money. We would still be well. We don't need to hide it, or flaunt it, or cause any other kind of problems for our competitors.

Richard Campbell: Well, and this is a karma thing too. One of these days they all are going to find a weakness like that and you'd like that phone call too.

Dana Epp: That's right, that's right.

Richard Campbell: Yeah. I do think that the security pros are grossly outnumbered and you've got to band together.

Dana Epp: Yeah, that's very true.

Richard Campbell: Yeah, it's an interesting problem. From a development perspective, I mean we're changing gears a little bit here around this, so now how many seats did you have out there? Is there something you've got to push out to everybody?

Dana Epp: Yeah. So our methodology works that we take advantage of IT automation systems like Active Directory Software distribution policies, Systems Center software packages, we've taken advantage of to see deployment agents and so when they get this as well as the standalone XE, we include a MSI and a special MSI builder that will create basically an update pack and they can just draw up in that in the Active Directory and it will automatically push out to everybody and update it real-time. The only issue is Windows Server 2003, unlike our credential providers on Windows Server 2008 and Windows 7.0 and things like that, Windows Server 2003 requires reboot because the Winlogon process only loads the code during boot.

Richard Campbell: Ugh.

Dana Epp: So that means that everyone has got to schedule this on the way so that it's in non-business time.

Greg Hughes: Sure.

Dana Epp: The good thing is essentially just like Systems Center I can look for that. I can tell when there's an idle time, when no one is on the system and they can push it out at that point and enforce it and they can even setup edging timeouts so I can say I need this to be pushed out within the next 48 hours and if it can't it will alert you to manually go and beat the guy and say, "Look, it's time to reboot. Will you reboot please."

Greg Hughes: Right.

Richard Campbell: Right.

Dana Epp: So that works out really well and with a lot of it. We work a lot with managed server providers and a lot of them actually -- these are guys that manage 500 small businesses and they're managing some 3,000 endpoints and so they can't

use things like Systems Center because those were not design for 500 separate domains.

Richard Campbell: Right.

Dana Epp: That's where companies like Kasia have these tools. They basically provide a script that can do that. So when we deployed out the fix, we included prescriptive guidance on how to use these tools to do that. So we've included a WMI scripts that can force the call to carry it down. We have a script that walks it, dumps the MSI to the endpoint and run it so that whatever the environment maybe, if it's Enable or if it's LabTag or all of these other vendors, they can all take advantage of using that same script set and push it out. In many cases, a lot of these guys, they only have a couple of hundred machines that this might affect but the nice thing is it will only take them 15, 20 minutes to schedule it and get it updated.

Richard Campbell: This still gets back to the question of this was not an exploit.

Dana Epp: No.

Richard Campbell: This had not been found in the wild. Was it really worth pushing a hot fix that fast? Can't you just put it into a point rev?

Dana Epp: So the way that we measure risk ultimately for our code is it's going to be the potentiality. What is the chance that it can happen versus the criticality of damaged potential that will ultimately occur if something goes on here?

Greg Hughes: Right.

Dana Epp: At the end of the day under those particular conditions, if you use the Windows Credential Manager, you use a console that will log on as administrator, you will bypass two effective authentication. You still need to know the administrator credentials so it's not like someone could remotely just get into the system, but the problem is that you are bypassing the auto-ability components of being able to say you are actually a valid person authorized to log in here. That to us is the highest risk you could possibly have. You have a potential to get on to a system without auto-ability control. So for some of our clients that have to meet compliance objectives, they cannot have a scenario like that so we have to provide a way to fix that. Ultimately to that though is we knew that there was a problem that was here, why would we want to hold off and fix it. You know, we use so much test automation here that it doesn't take us a lot of time to validate a fix, and because we use agile coding development here all the times we're always going to ship on those dates so at the end of the day once we go -- we made a shift here, we obviously moved our current sprint around and said look guys, we're stopping everything. Let's review this right here. Let's take a look at that thread model, let's take a look at the task plan of what we have to validate. Okay, can we build an automation test for that? Yes. Great. You guys build that. Let's go make this fix and once the fix is available run it against a set of tests and at that point we have a confidence that it's good. What would we wait for a point release? There's no reason to. We're just putting businesses at potential risk, and just because we don't know that no knows about it doesn't mean that it isn't a real exploitable vulnerability that somebody else may or may not know about. The good hackers will tell you that they have found a vulnerability in your system because they want to use it to their advantage.

Richard Campbell: Of course, yeah. Right. So you're saying you abandon a sprint that was currently under way?

Dana Epp: We didn't abandon it. We just paused it. We do 30-day sprints and in the process that we normally do that we know exactly we're in schedule. All we simply did was we, for that one week, we just basically adjusted the entire sprint at the sprint that we work on this particular thing and then we got back into the sprint and let it go. In fact, it's kind of funny because a couple of guys they weren't needed, but once I got involved there wasn't really a lot that these developers could do on that because there would just be conflicting developments so what ended up happening in that scenario is they went back into their sprint to do their bits. There's no reason for them to wait until they were needed. What ended up happening in that scenario was it allowed everyone to keep working. What's kind of funny is we actually were ahead in our sprint anyway so we looked like we're coming near, you know, we're about a week away, the development of sprint is done. The testing sprint is in place right now. The way we do our development, we did two weeks of dev, we do a week of test and automation to that and the review before it gets out there so what ends up happening at the end of that is we are still on target for all of our sprint items. So wish for it to be okay.

Richard Campbell: But you've already shipped this fix now.

Dana Epp: Uh-hmm. Yeah, it went out on Monday. No, last Thursday actually.

Richard Campbell: So you sort of did a mini one week sprint and shipped the release and now you're in sort -- did you sort of go back to before the sprint in terms of the source you're working with?

Dana Epp: So the scenario with this one, this particular code segment was not in this sprint.

Richard Campbell: Okay.

Dana Epp: So what's nice is that we wrote a stable code. I think the last bit that this code has worked on was several cycles ago. So it wasn't a big issue to bring that in. I will tell you that we basically abandoned our standard development methodology for doing a full sprint review because remember what we do is in day one we're obviously looking through the -- we're looking at the backlog items, we're taking an idea of what needs to be done, we'll assign that in there, then we'd spend half a day updating a thread model to whatever it is there and updating any other documentation that would be reflective of what's about to happen and then they go to do two weeks of your desk sprint, and then one week of your test sprint, and then you've got one week of your review and fix sprint, and then in the last day we do our spring review and then from there we can recycle it over and we spend half a day looking at say what was corrupt, what we do really well, what can we do differently and then we're back in say next month. What happened in this scenario is as this occurred, we couldn't do our standard sprint in a one week period and we didn't know how long it would take us to fix this because we had to really get an understanding of what was going on. Well, a lot of that was, for the first three days, was hurry up and wait. We were assigned to go to Microsoft support channels to get what we needed to so we have enough here, the time. People could still work on what they needed to get done and there was only like two days of, you know, everyone was working the four days to code it. Because we actually coded three different possible fixes because we did two based on our own knowledge of what we could do because we found a way to bypass -- once we found what was the root cause for the policy changes, we found a way to change the policy but in testing of that we had found, well, you could bypass policy by remotely changing the setting. Oh, okay. Well, that's not a good thing. We've got to look deeper and now it's at a point when they bought me in and said, "Okay, Dana, what are we going to do here," and I was like, "Okay, well, ultimately what's the call?" So we just basically use IE Pro, we took a look to find out what were the API calls being used. We found the specific call Microsoft's code was calling. We found a way to map it and we did and then with a little ingenious C programming with some point manipulation and the rest is history.

Richard Campbell: It does feel like you were worried about how long it was going to take for Microsoft to respond to you and were finding...

Dana Epp: I didn't know how big of a problem this was, how deep was the rabbit hole.

Richard Campbell: Yeah. What have I stepped in?

Dana Epp: Yeah. Well, we were just like okay, what other policy systems are out there? What other things can affect this? What don't we know? And that for me, you know, I almost take these things personally and it's not that I should because it's not like this is my problem but this is ultimately a company problem, it's my client's problem and so it's important that we get it right and so I know I could go pick up the phone and call some other guys in the security team and get some deeper knowledge in that stuff, but the problem with that scenario, first off you burn bridges with friends and you don't need to do that.

Richard Campbell: Yeah.

Dana Epp: But the second thing is that ultimately at the end of the day we needed to have a supportable mechanism for what we're going to do and that means we had to go through the proper support channels and what I was worried about and may actually start to go in that way and I was just happy that the right guys at Microsoft picked it up was when you get to tier one they have no clue. This is so deep beyond. They don't even know the API calls. They couldn't even type it out when they're writing it down and you're like trying to spell it out to them and you're realizing this is not a very good use of my time but it is something I have to do and I have to go through this pain and then get to the next level. The problem there is I didn't know how long it would take us to get there and where this thing is going, someone is finding themselves. What if this customer goes and tells someone else and they go look deeper? What could this do?

Richard Campbell: Right.

Dana Epp: At the end of the day we just want to make sure it was done right and I think we did a good job. You know, I'm very proud of my team. They had the luxury of having all the automation to make their life easier. So when I'm screaming at them to reprogram this situation, recode this condition, it's a matter of reverting the snapshot, rerunning against that scenario, reporting it back to me and then we can make a proper informed decision on what the heck is actually going on.

Richard Campbell: Well, Dana, a very cool story. Thanks so much for sharing it with us.

Dana Epp: No problem. It's kind of funny now that everything has calmed down, everything is out. It's nice to look back and just say what did we just do? And from triage to fix to release, these kinds of things they consume a lot of time but when done right everyone wins and that's a good thing.

Richard Campbell: My contact with this whole thing was you and I were talking about doing another show and in the middle of planning that you said, "Sorry, I have to put everything on hold. We're dealing with a security crisis." and then a few days go by and you're like, "Okay, it's over and we've got the

handle. We should just talk about that. Let's talk about that." And here we are.

Dana Epp: There you go.

Richard Campbell: Dana, thanks so much for coming on the show.

Greg Hughes: Thanks, Dana.

Dana Epp: No problem. Thanks for inviting me.

Richard Campbell: And we'll talk to you next week on RunAs Radio.