IBM

Moderator: Steve Southworth
November 15, 2007
11:00 am CT

Operator: Good afternoon. My name is (Robin) and I will be your conference operator today.

At this time I would like to welcome everyone to the Security Best Practices with IBM Rational Conference Call.

All lines have been placed on mute to prevent any background noise.

After the speaker remarks there will be a question and answer session. If you would like to ask a question during this time, simply press star then the number 1 on your telephone keypad. If you would like to withdraw your question, press the pound key.

Thank you.

Mr. Southworth, you may begin your conference.

Steve Southworth: Hello and welcome to Rational Talks to You Telecon series Security Best Practices with IBM Rational.

I'm Steve Southworth with IBM's ISV and Developer Relations and I'll be your host for today's call.

Today's topic will cover learning about application testing best practices from Watchfire, an IBM company. You'll learn about Web application security and the shifting focus from network base to application base security programs.

As more and more hackers target vulnerabilities at the application layer, many organizations find they are left exposed as they struggle to effectively combat this growing threat. We have two excellent speakers for today's topic, David Grant and (Brian Blackshaw).

As Watchfire's Marketing Vice President, David is responsibility for global marketing and product management, which includes overall product strategy and all facets of marketing communications. David brings 12 years of software industry marketing experience and expertise.

(Brian) is Watchfire's Director of Product Management and is responsible for product definition and strategy for the security product line. (Brian) has been working in the software industry for 15 years focusing on the security space, most recently at McAfee Security and Cisco Systems.

You won't find any slides for this teleconference. We want to keep it fast moving and very interactive. In fact during this presentation we will frequently open up the call for an interactive Q&A session. These calls are really for you to get your questions answered by our expert panelist. So write those questions down as you think of them and jump in.

So let's get started and I'll turn the call over to David and (Brian).

(Brian Blackshaw): Thank you, Steve. This is (Brian) speaking and on behalf of Dave and myself we're really excited to be part of this. We understand it's the first of this type in the series. And part of it is a lot of interaction. So we definitely

encourage a lot of QA throughout. Be sure to check in after each subject to give you a chance for questions. So we really encourage that.

As Steve mentioned, the Web application security is the focus of our talk today. So (Dave), could you give us an overview of the topic of Web application security?

David Grant: Sure. Thanks. And again, thanks everybody for joining us.

So when you think about application security a lot of people have different definitions for what exactly that means. You could - some people think it's more about single (scion) or access management to the applications. But we're really talking about when we talk about application security is ensuring that any business applications or B to C consumer oriented Web sites and applications that are delivered to a population are not vulnerable to malicious individuals and hackers.

What we've seen over the past, you know, two to three years is that applications themselves are very vulnerable in general. And the reason being that, you know, most people who build applications and Web sites just don't need to think or are not trained to think about security. They're more thinking about functionality, time to market, reducing costs associated with building these applications.

And by nature if you think about an application, especially one that is, you know, externally facing to customers and partners and potentially employees, they are meant to interact with people. They're meant to provide self-service. They're meant for people to gain access to a company's data and information.

But the problem is that by definition because they are built that way it is very difficult for those applications to understand and know the difference between somebody who is, you know, reputable and is a customer potentially from those that are trying to do something malicious like a hacker. And that is really what the topic of application security is all about. It's about ensuring that those applications that you are deploying - building are secured.

And if you are in a development organization or security organization, this is probably something that has come on your radar in the last year or two depending on - for a number of reasons. One is you can't pick up a newspaper any day of the week now and not see some latest data breach where people have been exposed.

And if you look at these data breaches, in the past 12 to 24 months we've seen a significant change in the direction of those breaches. Where as before they used to always come in through the network vulnerabilities in the areas, now the lion's share of these attacks are happening at the application layer.

Well, why is that? It's because typically applications are pretty vulnerable for a number of reasons, hackers know that and they know that is the place that they should focus their attention on, not to mention that applications typically collect a lot of very sensitive information about customers and partners and employees. So they're a great target for these people.

And so hackers have turned their attention applications because they tend to be a weaker link in the IT infrastructure. They're an attractive target according to Symantec, Gartner Group and our own internal research here at Watchfire. We've seen that most Web sites in general - somewhere - anywhere from 70% to 90% of applications we come across are indeed vulnerable to security issues.

So that's what we're going to talk about today is application security. And what I'll do is I'll turn it over to (Brian). And I'll ask (Brian) to maybe clarify for everybody - a lot of the misconception we hear from people is, "Well, what is the difference between application security and all the other security I hear? I've already heard about vulnerabilities at the network or at firewalls." Maybe you can talk about how we're dif- how it's different and then maybe give us some examples of what an actual Web application vulnerability exploit would be on a typical application.

(Brian Blackshaw): Sure thing. Sounds good. I'll do that and then we'll be sure to open the lines for questions after that.

But in the meantime if you want to imagine and end user interacting with a Web application, during that interaction the HTTP traffic is going to be traversing over a number of security elements. You know, as Dave mentioned, firewall, intrusion detection, there might be some encryption involved. Typically all these elements are dealing with the traffic in real time while it's in the data flow.

So you'll have a firewall, for example. It's going to be enforcing a policy, i.e.; what ports are accessible, the IP addresses that are accessible; an intrusion detection or intrusion prevention device that's going to defending against a certain set of signature based attacks or known attacks. Again there might be some encryption that's also occurring to the traffic while it's in line. And again, it's all dealing with traffic while it's in the data flow.

And these are all very important elements in securing, you know, an enterprise. As Dave mentioned, large investments have been made in securing

the perimeter. I think we have spending data that says about 90 - or statistics that say about 90% of security spending is in securing the perimeter.

But much like the sort of antibiotic effect, malicious attackers are going to be finding, you know, another way in. So while the traffic flow to the Web application is typically well secured at the perimeter, it's important to note that the Web application its self typically has a trusted relationship with the backend systems. So this is, you know, the backend servers, the database and it's this trusted relationship that a Web application security exploit is really attempting to leverage.

So the execution of such an attack isn't necessarily sending, you know, malicious code or a virus or anything like that to the Web app, it's more actually interacting with the Web application through perfectly viable methods of communications. So you'll be over port 80 or port 443, the HTTP traffic isn't going to be triggering any signatures on an IDS machine.

But at the same time you are interacting with the parameters of the target Web application and finding out a little bit more. For example, some error messages might give you some clues as to what the SQL syntax is of your database. And, you know, that could result in, you know, actually gaining access to that information without authorization.

So what a Web application scanner does - it does not actually look at traffic in the data flow. What we do is we actually scan the application, we crawl through the entire application and we try to enumerate all the parameters that are associated with that application. This would include queries strings, URL paths, forms or cookies. And we basically perform a large number of tests against these parameters. And based on the response of the test we can

identify whether or not your application is actually vulnerable to the - to a Web application security exploit.

And as Dave mentioned, in many cases it is and what we're dealing with a lot of times is just the lack of awareness of a developer when they're actually coding the application. They might not realize the implications of not sanitizing the input when the ap- they're dealing with time to market pressures, the apps security testing might not be part of the QA process or the staging process. So that is how actually such vulnerable applications often make it to the field.

So there are a couple more network elements that I wanted to talk about. So the behavior of the Web application scanner probably most - is closely associated with a network vulnerability scanner such as Nessus. So - and we often get asked, you know, "What's the difference between the two?" And it's actually a pretty big difference.

Both are scanning, you know, environments. We're scanning Web applications and the network vulnerability scanner is a scanning environment for hosts and trying to identify, you know, the network devices, the hosts, the third-party applications across the entire enterprise network topology. And identifies what vulnerabilities exist. And these are typically known vulnerabilities or misconfigurations.

Where as what our scanner is doing is actually interacting with the application and identifying unknown vulnerabilities. So it's more of flaw in the application as opposed to a known vulnerability. And that's sort of the main difference between those two products. Excuse me I'm retaining a sneeze here.

The other network element that I wanted to talk about is a application firewall. Now an application firewall is actually trying to identify application vulnerabilities while in the data flow. And you probably imagine that the challenge is trying to identify what good traffic is versus bad traffic given, again, many exploits are executed through perfectly viable traffic over port 80 and port 443.

So the configuration challenges around a Web application security - sorry - Web application firewall are pretty high. And I think that's probably why we haven't observed a, you know, a large deployment of such network elements in the past.

In fact Watchfire originally had one and we were selling one but we actually sold off that part of the business because it just didn't seem to be much of a proactive approach. Where as the scanner its self, if you are scanning either, you know, during development, during testing or at staging, it's a far more proactive technique to make sure that there are no Web application security vulnerabilities in the underlying app.

So, operator, maybe I can get you to open up the floor for questions?

Operator: At this time I would like to remind everyone in order to ask a question, please press star then the number 1 on your telephone keypad.

We'll pause for just a moment to compile the Q&A roster.

Again, to ask a question, please press star 1.

(Brian Blackshaw): Maybe in the mean time I'll ask Dave, you know, PCI Compliance is quite a - is getting quite a lot of press and attention. Dave, can you comment on why PCI Compliance is important for Web application security?

David Grant: Sure. It's probably the most relevant regulatory issue right now that most companies are dealing with. So for those that are on the call today you might have heard of this before, you may not have. But the Payment Card Industry Standards, which PCI stands for, is VISA, MasterCard and all the other major credit card hole- companies attempt to mandate to anybody that uses credit card information on their Web application.

So any Web application at all that collects credit card information from any of those credit card companies has to go through a set of rigorous Web application security testing steps in order to be - to pass their certification. And, well, what does that mean? It clearly says that you have to test those apps on a periodic basis for vulnerabilities and make sure they're not vulnerable.

If you don't comply with that regulation and you - they either find out by they're own self-service audits or there is potentially a worse case scenario, there's a breach in your organization, they can levy pretty severe fines as well as take back the right for you to use those credit cards. And for a lot of companies that could be a crippling event to happen to their business. So we're seeing a lot of momentum building for application security because of the Payment Card Industry Standards.

(Brian Blackshaw): Great.

Operator, did any questions come into the queue?

Operator: There are no questions at this time.

(Brian Blackshaw): Okay.

David Grant: Well, this - (Brian), there's one other thing. I know there's one coming on email here and it asks the question of, you know, does the fact that, you know, Web 2.0 technologies - does that have any impact on what we're talking about today?

(Brian Blackshaw): It absolutely does. You know, Web 2.0 primarily refers to the different techniques in which you can create a Web application. It's often, you know, rich Internet applications that are using AJAX for example. The technology of AJAX and the fact that a lot more is happening just on the client side impacts how you can identify Web application security vulnerability.

So what you need is a scanner that is able to actually not only automatically crawl through this AJAX app, you know, executing XML HTTP requests which, you know, is sometimes a challenging thing to do. But you also - you have to be able to identify specific exploits now that are targeted directly towards that AJAX application. So that's an important piece of criteria when determining what type of scanner you should buy.

And same time Web services is becoming, you know, an increasing, you know, use with SOA. We now have applications which would typically be a monolithic application being broken down into much smaller components. And the method of interacting with those smaller components can - you basically have dramatically increased the surface area of the possible attacks now when you've broken that application up. So it's very important to include Web application security on either an app that's being interacted with by a HTTP purely or might be by (soap) over HTTP in the case of a SOA app.

(Brian Blackshaw): Perfect. Yeah, I think just the fact that there can be a lot more interactivity and sharing of these components as well obviously will impact security.

David Grant: Right. Yeah, because before in a large application where the method of interacting with that application might have just been the input parameters. Now the API calls and the methods of interacting with the application between the systems are new points of entry to the application that really needs to be scanned as part of your security test.

(Brian Blackshaw): So, Dave, you know, not to date you but you've been in this industry for quite some time now. Can you maybe comment a little bit on how you've seen enterprises and how they've addressed Web application security? How you've seen that change revolve over time?

David Grant: Sure. And I think, you know, the timing of, you know, IBM acquiring Watchfire, our organization is - indicates exactly what we have seen in the trend of security testing. So I'll try and tie it all together.

But, you know, if you think about the topic of application security, it's relatively a new one in the whole field of IT security. It's not that new. You know, we've been doing this for almost ten years. But the fact that Web applications aren't new either but really from a business process they - you would consider them relatively new.

They have only been really around collecting sensitive information in a very dynamic fashion say for five, six, seven years. So, you know, we've seen people go through different phases to address this problem. You know, there's still lots of people that don't do anything for this -- absolutely nothing. Their developers and the quality assurance teams don't do any testing. The security

teams don't do any auditing. And unfortunately a lot of times they're the company that you'll see in the press that will have breaches.

You know, hackers are getting very targeted and smart as you've heard from reading articles. It's all (driven) by organized crime now. This isn't just people having a bit of fun trying to take down Web sites and throw up a picture on a Web site, as we saw in, you know, late '99 - 2000 era.

Now it's actually out for profit. So it's very targeted. They will troll the Internet and find and explore and look for the weakest link on these various Web sites. So I hope that most people are beyond the doing nothing phase. But that's obviously where people start.

The second phase that we start to see how people address this problem is they will go out and hire consultants. They will find people who they call, as most are known, as ethical hackers, or penetration testers is another name. And they'll pay these people to come into their environment and look at their code, try and hack into the application and show their development team that they can gain access to database tables and other things via the application. And that's a consultant. And that can range anywhere from, you know, $10,000 all the way up to $50,000 per application per time tested. So that's one way of doing it.

I think what typically then happens with most organizations is they recognize that this is something they need to pay attention to. They know consultants and having outside perspective is always going to be important on a good compliment to any IT security plan. But they also recognize this as something that has to be brought into their own organization - they - for a number of reasons. You know, definitely the cost prohibits you from doing frequent tests with consultants, so if you want to do frequency issues.

So really the other way of solving this problem is to test applications throughout the software development lifecycle and you use products that are called application vulnerability testers or scanners, such as those offered by now Rational AppScan. And what you do is you point these automated testers at the application whether it's at a component of an application or an entirely build application. And it comes back and automates the test and comes back and tells you everywhere where there's vulnerabilities.

And typically how people progress through this maturity is IT security are the first group to say, "This is something we have to do." And so they go out and buy a Web application tester product like AppScan. And they'll run tests against the application later on in the software development lifecycle. So development will already have coded and built the app, QA would have done the functional and regression testing and other things and then it'll get almost ready to go live.

But before it goes live, a security team, who really are a SWAT team for handling these problems, will scan the application with Rational AppScan or other products. And they will say, "Here's a lot of security issues. I'm going to send this back to development and quality assurance and make sure they fix this. I'm not willing to accept this application's risks because of the vulnerabilities." And that's typically how a lot of people today are handling this problem.

Now obviously like any other problem, I hope most on the phone today will recognize the benefit of not waiting that late in the software development lifecycle. You can't wait until you're ready to flip the switch and put this out into the live environment. You should be testing these things earlier and as soon as possible in the development lifecycle.

So the most practical enterprises out there today are actually building in checkpoints along the software development lifecycle. So in development, in quality assurance and in the security realm they're using IBM Rational AppScan to test their applications or pieces of their applications for these issues.

And that is the best of br- you know, the best of breed approach to this problem because if you're isolating this testing into one small group in IT security, it's just impossible for IT security to keep up with the amount of volume, of changes in applications. We all know that applications (unintelligible) development is, you know, they're changing daily, weekly, monthly. It's no longer, you know, these build cycles for wet applications that are, you know, monthly or quarterly. They're a lot shorter.

o these IT security teams can't keep up and so they need to move it back further in the lifecycle, not to mention we've all heard the stats that have been around for years that it's a lot cheaper to catch these problems early in the software development lifecycle than it is to wait until production. So that's some of the best practices we're seeing - people moving throughout the software development lifecycle.

You know, obviously that is why the Rational team decided to go out and look for technology to acquire to compliment their existing quality management portfolio. And so Rational AppScan now provides the security component of their quality management portfolio. And it really helps us build into that software development lifecycle.

Brian), do we want to see if there's any questions? I know there's a few coming in on email. But I will let the operator ask some live first if they want

and then if not I'll go to some on the email here and then we'll move onto the next topic.

Operator: At this time…

David Grant: Operator?

Operator: …I would like to remind everyone in order to ask a question, please press star 1 on your telephone keypad.

We'll pause for just a moment to compile the Q&A roster.

Your first question comes from the line of (Shannon Dugall).

(Shannon Dugall): Yes. Hi. I had a question. You were talking about testing applications throughout the software development lifecycle. Does that mean that application testing products integrate with the development tools?

(Brian Blackshaw): Yeah, I mean that's an important of try- you know, Dave, was talking a lot about trying to get this application testing being performed earlier in the software development lifecycle and in order to do that you want to make sure that it's happening in a rather unobtrusive way.

I mean we've already commented on a developer, for example, dealing with their own time to market issues. And unfortunately there seems to be, you know, a lack of education even in standard, you know, college and university curriculums around coding on the topic of security. So often it's a case of making sure that you're building awareness, for example, early within the software development lifecycle, but also equally important is making sure that you're interoperating with the tools that they're already using.

What we found is, you know, we - the - you want to work in - fit into the work flow of the developer. So, you know, what we're working on right now is creating an integration with the Eclipse environment, for example. So we will essentially have a RAD plugin that allows a application security test to be invoked directly from their IDE. And it makes it far easier just to include as part of your coding process, you know, testing the pages that you're developing and the parameters for security.

At the same time in the testing side of things, you want to make sure that you're interacting with the testing tools that are being used by the QA professional. So this might be the ClearQuest defect tracking system, for example. And what we want to - what we can do currently in our product is be able to push security issues that's found by one of products and just push that directly as a security issue into the ClearQuest defect tracking system. And then again, it becomes part of the overall work flow and interaction between the QA department as well as the developer. And we find that's the most, you know, efficient way of trying to build in Web application security testing earlier in the SDLC.

At the same time we also see the infosec person, who may be responsible for actually mandating the testing of these issues, still wants to maintain some oversight - what kind of testing activity is going on. They might also have a test policy, for example, that they're trying to enforce and so they might actually be creating that test policy and then pushing it down to the developer or the QA. So, you know, we have some enterprise tools as well that allow you to interact with the actually endpoints that a QA professional or a developer might be using.

(Shannon Dugall): Great. Thank you.

Operator: Again, if you would like to ask a question, please press star 1 on your telephone keypad.

David Grant: Well, I think, (Brian), that was a good segue into - you know, I think a lot of people on the phone today hopefully, you know, are developers themselves or develop managers for quality assurance folks.

Why don't you - I know you probably have the most experience at Watchfire and with AppScan of helping companies, you know, expand security tests into the SDLC. Why don't you build out what I talked about and maybe talk how do you actually use testing products like Rational AppScan to integrate within all those various phases whether it's development, build, QA? How - what are the best practices we usually talk about when we talk to clients who are successful in this area?

(Brian Blackshaw): Sure. Yeah, I think that really sort of focuses on not trying to, you know - a penetration tester, for example, or even the infosec professional, is going to have a very broad set of tests and a very thorough method of testing all aspects of the application and the infrastructure. Trying to take that type of comprehensive testing and just move it into the SDLC as is not a recipe for success as we've experienced, or observed at least, with some of our customers.

What we now - we go to our customers with is some best practices that really focus the developer and/or QA professionals testing just to make sure that, you know, in combination of building awareness, you know, our tools include some Web based training and remediation information that really helped translate the sort of securities speak into, you know, "Hey, you just have to

sanitize input. This input parameter is not being sanitized and that could be - result in bad things."

So there's the awareness that we try and build and that's got to be part of the process when you're bringing the SDLC deeper - or the testing deeper into the SDLC. At the same time you really want to focus your test policy. You want to make sure that you're at this stage of game just looking for the hanging fruit.

We don't the developer or the QA trying to access, you know, whether one issue might be more important than the other or making the QA professional, who's not a security expert, trying to defend to the developer, for example, why something is actually a security issue. It's the, you know, defense of that of being a security issue versus on the functional side where it's far easier to say, "Hey, this isn't working. I can tell because, you know, it's not letting me input something or there's a fundamental quality problem on the security side."

You know, it's just far harder to enforce something that might be more of a medium priority for the developers. So again, the notion is to make sure awareness, and that's through remediation information, that's through Web based training, make sure that's a component of your implementation; focus the test policy; make sure you're really just looking for the low hanging fruit.

We have a lot of infosec person - people who are mandating Web application security within an enterprise just saying, "Hey, if I can remove cross sight scripting and sequel injection errors for my Web applications, by the time it gets to me, I am in far better shape than I used to be." So start with that and then as these people become more subject matter experts and security - you're

more comfortable with the security issues, you can broaden the test policy. But there's really now reason to rush that in the beginning.

And then finally the third component would really be making sure that the tools that you're deploying interoperate seamlessly with the tools that those professionals are already using, whether that's the IDE, the defect tracking system, the test management system, making sure that you play well with those components is important in a successful deployment.

David Grant: And how important would you say training is in part of all of this and what's the best - what's best practice there?

(Brian Blackshaw): Training and the awareness - the Web based training that I mentioned is really popular actually. It helps a lot. Again, it's unfortunately security and security - secure code best practices aren't taught enough in existing curriculums at least. So really the only way that some of these developers are learning how to code with security in mind is sometimes through our tools. So the Web based training is very important.

And what we've also found is it's not, you know, send your developer to a week long course or a day long course or even make them watch an hour long session. What we've done is in our Web based training is, you know, focus the training down into really bite size pieces as we call them. So they can range anywhere between two minutes, five minutes, ten minutes but these are really short training courses just so you can say, "Hey, how do we remediate, for example, a sequel injection error?" Well, here's a three minute tutorial on not only what it is but how you can make sure you don't make it happen - don't let it happen again.

And in fact our fix advisories - when you are interacting with our product on IBM Rational AppScan, you have a fix advisory which gives you information on the vulnerability, how to remediate it and then also remediation view. We're including now hyperlinks into these fix advisories so that you can click on that and get a three minute overview on what the vulnerability is and how you'd remedy it.

David Grant: Perfect. Do we want to see if there's any questions? It's been a quiet audience today. So I know there's been a couple more come in on email but I'm going to save some of those until the end because I think they're more generic things I would like (clear) up with.

So if there is any questions at this point, operator, on the SGLC and how to get development involved in application security testing, maybe you can ask now but I think everybody knows how to ask the questions. So if you want to ask, feel free.

Operator: At this time I would like to remind everyone in order to ask a question, please press star 1 on your telephone keypad.

Your first question comes from the line of (Shannon Dugall).

(Shannon Dugall): Yes. We outsource the development of our applications. What does that mean for security?

David Grant: Good question and one that comes up all the time lately because outsource can mean a lot of things. It can mean I buy packaged applications from vendors, I buy packaged applications from consultants, I outsource it to overseas operations.

There's a lot of different ways you can outsource application security and unfortunately a lot of companies sometimes think that because of that, you know, the only responsibility is put on the person providing the application. I think what we've seen with our clients is probably the opposite in terms of you've now outsourced a pretty critical function or important piece of your business and how comfortable are you that that organization has properly secured that environment before you take ownership of it.

So a lot of clients these days are asking, you know, AppScan and Watchfire as part of Rational to actually help them build out (SLA) or software, you know, license agreements where they actually - service (unintelligible) agreements - sorry - where they actually spell out that these outsourcers have to provide some due diligence around what they did to perform security audits of those applications.

So I really like to make sure everybody comfortable and knows that just because you outsource some of your application or all of your application development, that is by no means a safety net for these issues that we've been spending time on. And in our experience actually sometimes it makes the problem worse because you have outsourced this for cost reasons and a lot of those vendors that are actually out there and consultants that are building those applications don't have the skill set or the investment required to do this extra diligence around security.

So I'd encourage you if you do outsource your development that you talk to your outsource company that you're using and you ask them what are they doing for security, what kind of skill set do they have, especially if these are high risk applications that you are acquiring from these companies.

(Brian), will you add anything or did I cover most of that?

(Brian Blackshaw): No, I think you covered it well, Dave. I mean it - like you said, I mean these applications are coming typically from the outsource company with no SLA around security. So, you know, I would stress again how important it is to make sure that any outsourced application is, you know, - Web application security testing has to be part of the staging process before that app is put live.

Operator: Again, to ask a question, please press star 1 on your telephone keypad.

There are no further questions.

(Brian Blackshaw): Okay. Well, I think, you know, - Dave, I'm not sure if there's anything you wanted to add? We certainly appreciate the time.

And we'd like to stress again, you know, what we've seen in the Web application security environment - (Meter), for example, who publishes the CBEs, has, you know, indicated that cross sight scripting and sequel injection both Web application security specific vulnerabilities have surpassed buffer overflow of the tried and true buffer overload - overflow technique as a method of exploit. So we truly do think it's important that as part of your process Web application security be built in before these apps go live. They are the new target.

David Grant: I guess one of the things I would like to quickly add as well is that a lot of development organizations think, "Well, IT security are responsible for security so why can't they just handle this problem?"

Hopefully we've done a good enough job describing that this is a problem that is too costly to wait until IT security and IT security are security experts but they don't know the application like a developer and a quality assurance

professional know the application and architects. So it really - this security problem more than any other IT security problem it does require knowledge of the application, how's it architected, how it works, what it's goals are and that's what we encourage, not to mention the cost benefits and all the coverage elements that we talked about from using - from embedding it into the software development lifecycle.

The good news is there's lots of automated tools out there to help you with this problem. Rational has a full portfolio now with AppScan to embed within all of their stages of the Rational process for developing applications and delivering those applications. So you do have automation and support there. So hopefully it's something that you can explore on your own time.

(Brian Blackshaw): Yeah, I would add to that point, David. Your - it's an important point that bring up about IT security being the sole person responsible for Web app security. You know, that IT security person has typically also been small in numbers, even if it's a large perimeter that they're trying to secure. It's usually large scale firewalls, maybe a large number of firewalls but it's all something that can be managed typically with a proportionately small number of infosec security professionals.

However, when the scanning and testing of a Web app now becomes necessary what, you know, these IT security professionals are now finding is that they can quickly become a bottleneck if it happens to be an enterprise which is active and publishing or changing Web apps in general. And that is why they're coming to us, not with, "Hey, prove to me that Web application security is a problem," but, "Let me know how I can operationalize this within my enterprise."

And that's where we thought, you know, bring to them the tools that Dave was referring to, AppScan, IBM Rational AppScan, AppScan enterprise, all tools that help get other people involved - non-security experts involved in the process of mitigating these types of issues.

David Grant: Excellent. Well, operator, one last chance for anybody out there to ask any of the questions of myself or (Brian) on the topic of application security.

Operator: Again, if you want to ask a question at this time, simply press star then the number 1 on your telephone keypad.

We'll pause for just a moment to compile the Q&A roster.

Again, to ask to ask a question, please press star 1.

David Grant: Hopefully this is because we've been so clear today, (Brian), and it isn't the lack of interest in the topic because, you know, hopefully everybody that's involved in application development would be interested in this topic.

Steve, maybe I'll turn it back over to you to conclude the session.

Steve Southworth: Hey, thank you very much. Thank you, David and (Brian). That was really very interesting and a very valuable session. We appreciate you taking the time to share your knowledge and your experience and this is a very, very important topic - Web application security. So thank you for taking the time out.

If you'd like to listen to this conference again or share it with your colleagues, this will be made available for replay in MP3 format in about 48 hours. Check out the Rational Talks To You page at ibm.com/software/rational/talks. This

link also includes the other titles in the series, so mark your calendar and be sure to register for our next talk on Thursday, November 29, Grady Booch on architecture at 1:00 pm Eastern.

And again I'd like to thank our speakers, David Grant and (Brian Blackshaw) for being with us today to talk about security best practices with IBM Rational. And we'd also like to thank you our audience for your interest in IBM.

We hope to see you back for another one of our events in the near future. So thank you very much everyone. This concludes this session.

END