

The Deep Queue

WebSphere MQ Security Podcast
Episode #10, May 2, 2009
Cash in on mortgaged risk!



Welcome back to The Deep Queue! This is Episode #10 for May 2nd, 2009. The podcast is about a week late this time because I've been preparing for the IMPACT conference which is next week as I'm recording this. I'll be delivering three WebSphere MQ security presentations this year: Basic WMQ security, which is the presentation formerly known as Hardening WebSphere MQ; Advanced WebSphere MQ Security; and Introduction to WebSphere MQ Extended Security Edition.

Thanks to Arjan Van Vught and Oliver Fisse, I will be able to do live demos at the two MQ security presentations. Arjan rewrote my WMQ Connection Exerciser as a stand-alone program so I can run it on my laptop. The version I wrote was imprisoned in its development environment and I didn't want to install all of that on my laptop. Oliver Fisse wrote a penetration test toolkit which I will demonstrate in the Advanced Security session. Sorry, neither of these tools are approved for external distribution and I can't give them out. I get that question a lot.

Getting back to the topic of IMPACT, IBM is getting their money's worth out of me this year and in addition to the MQ security presentations I will also be at the Business Partner Networking Reception, the Premium Zone Meet the Experts, several one-on-one BPM meetings, customer meetings and of course the WebSphere MQ Birds of a Feather. Whew! I'm exhausted just thinking about it. But this is what I most enjoy about my job. The conference allows me to reach a large audience but, unlike journal articles, also provides instant feedback and dialog. So it's possible to identify a disconnect between me and someone else, track it down, fix it and really come to an understanding. The security topics are complex and tediously detailed so this is the best way I have found to teach the subject. Even better, it's the best way for me to learn from other people. I always feel like I make a difference at the conference but the fact is I get way more from it than I give and that is because of all the people I get to talk with in presentations and especially one-on-one. So thanks in advance to all of the people attending IMPACT this year, you make all the hard work worthwhile.

Switching gears for a minute, I always start off the podcast with a humorous story and relate it back to security. This time you won't be laughing *with* me but you just might end up laughing *at* me. When I was just out of high school I bought a car for \$50. Yes, an actual working car for \$50. It was in pretty good shape, it kept me dry in wet weather and it got me from point A to point B. The problem was that my initiative for keeping the car maintained was in direct proportion to what I'd spent on it. Sure I replaced consumable things like gasoline and air in the tires, but an oil change represented an investment of about 25% of the cost of the car. Four of them would exceed the cost of the car!

So my approach to this problem was simply to ignore it. Then one day while driving down a very crowded divided highway with four lanes on either side, I noticed I was suddenly all alone on the road.



Many of the cars in front of me were outpacing me while the cars next to me and behind me had all dropped well behind me. It was a little hard to see the cars behind me because there was a lot of smoke coming from somewhere. Then when my car started to make a noise like a million fingernails on a million chalkboards I realized the smoke was coming from my car! I pulled off the road immediately, landing in the parking lot behind a grocery store. When the car stopped moving, the smoke quickly enveloped the car and I had to get upwind of it to keep from being overcome. I turned the motor off for the last time and caught the bus home.

Go ahead and laugh if you want, but I was a kid who learned best by experience and experiences have consequences. Take for example, the time I set the lawn mower too low. I thought it would save me a week of grass cutting. What it did was weaken the grass to the point where it allowed sandspurs to fill in and I spent the rest of the summer pulling them out and reseeding. Of course, there are consequences and then there are consequences. Some, like the lawn, were easily recovered from. Others, like the car, had considerably more impact to my life. Still others, like the explosion that took one of my fingers and nearly took my life, were catastrophic.

Eventually the thing I learned from an accumulation of these minor and major events was that I needed a way to learn other than from direct experience. I needed to learn to anticipate the consequences of my actions and to balance risk against reward. It's not that I stopped taking risk or using experiences to learn, but I at least avoided the risks that were well out of proportion to any possible reward. I stopped, for example, making homemade rockets out of metal cartridges with gunpowder propellant.

Most months you'd be wondering about this time what my story has to do with MQ security. But this time, I'm guessing you are seeing the link. It's about risk – and not just any risk but asymmetric risk where the consequences are far out of proportion to the rewards. I'm talking about the fact that most companies out there, probably even listeners of this podcast, are metaphorically making homemade rockets out of metal cartridges with gunpowder propellant.

Allow me to explain.

We get what we measure and in the corporate world what we measure is cost and time. Did the project complete under deadline? Was it under budget? Good! You get a bonus. But what constitutes “complete”? Again, it is those things we can measure. So the application is “complete” when all of the functional requirements are met. The application can accurately make credits and debits, it performs up to a thousand transactions per second, and it provides a variety of security roles for administrators and ordinary users.

But what about the things we don't measure? Suppose the project that completes on time because at the last minute, two weeks of planned testing were omitted? Or it was under budget because the security components in the DMZ were omitted? These actions exchange time or money for latent risk. Because the risk cannot be measured accurately, if at all, the incentives in the system tend to reward this kind of risk. If you have two project managers hypothetically working on the same application and one ditches the testing and the security hardware to come in on time and under budget while the other does not, the project manager who incurs the most risk on behalf of the company is the one who is rewarded.



This same thing happens at a macro level with the company itself. Every year in the name of efficiency, companies ask their departments to do more with less. Anything that is non-essential gets left by the wayside as costs are pared down to the bone. The problem is that what is considered “essential” are those things that can be measured, and of all the units of measurement, money is the one that is most important.

So if I add memory or another CPU to a server, the difference in throughput can be measured with precision. If a network administrator is fired and the work absorbed by the remaining team, the difference is a measurable cost saving. But after decades of this, most shops end up running with skeleton crews, working 50 or more hours a week, with so little time that work is selected on a triage basis. Nothing of strategic importance ever gets worked on because there is no time. Because there is no excess capacity on the team, it is impossible to absorb any new project without shelving some other project. And anything that does not produce an immediate, measurable benefit does not get resources assigned to it.

In this environment, security simply does not get funded. The costs can be measured, certainly. The benefits on the other hand are extremely hard to quantify. If you invest in that security gateway in the DMZ, how do you know it was worth it? How can you tell the difference between the security provisions stopping an attack versus the attack never happening? Or if you do detect the intrusion in progress, how do you quantify the loss that was prevented? That disparity between our ability to quantify the cost of security versus our inability to quantify the benefit means that if we ever do get funding to build something securely, it will be the exception rather than the rule. Later when comparing results, the project with security will be remembered as the one that was more expensive but not better in any measurable way than any other comparable project.

So day after day, year after year, decade after decade, companies have been steadily accumulating risk and the risk is asymmetric. It is far out of proportion to the cost. The problem is systemic and affects any kind of risk that is difficult to measure but MQ is the one I'm qualified to actually do anything about so that's where I'm going to focus here. In the case of WebSphere MQ there has never been a publicly reported breach in the 15 year history of the product. Had there been a breach, companies who invested in security could justify their expenditure. But there wasn't so companies that invested in security look worse in the market because they have nothing to show for it. Companies that didn't invest in security look the best because they have higher returns on their revenue. That delta, which could be a few tens of thousands of dollars in smaller shops or several million dollars in larger shops, represents the financial benefit of accepting the risk. But because the risk is asymmetric, a single breach could erase that fictitious financial gain tens or hundreds of times over.

Take for example the case of a bank that tried to recreate a production problem in a non-production environment. When they reran the day's transactions they ended up sending several billion Euros of duplicate transactions. Reconciling and correcting the problems cost well more than it would have cost to simply filter MQ channel connections by IP address or to firewall the production network, and that doesn't even begin to factor in their reputational losses. Had this been a hack and not an accidental breach, the results could have been catastrophic. It could have collapsed the bank.

Or consider the Heartland breach. This was not reported to be an MQ breach but it illustrates the point. There was a sniffer on the network collecting interesting data packets and shipping them to an overseas server. The breach depended on the ability to put a server's network interface card into promiscuous



mode to sniff packets and then to have an open route to the Internet to send the data out. Stopping it would have required the ability to scan the servers for the NIC configuration and report any in promiscuous mode or to set up the firewall to block external connections from the production network, especially if they go to anonymous overseas servers. If you just look at what the breach cost Heartland directly, and don't figure in what it cost all the banks and consumers, the loss dwarfs what it would have cost to prevent the loss. But the day before the breach was discovered, Heartland would have looked *better* than some other payment processor who had actually implemented these security measures. If you were an investor you would have been praising Heartland the day *before* the breach. The day after? Not so much.

So how do we fix this? My approach lately has been to start getting people to measure WebSphere MQ security. As I said earlier, you get what you measure. In the case of web applications, many of the attacks are well known and sophisticated vulnerability scanning tools are available. These tools make it possible to measure vulnerabilities and present the findings as a quantity that can be tracked over time. We know what the defect rate is today and whether it goes down tomorrow. The application must now not only come in on time and under budget, but also must be within some defect threshold (preferably approaching zero) in order for the project manager to get that bonus.

In addition to the existence of tools, there are also generally accepted audit standards and specifications that must be met in the web application world. These too are measurable. In this case, they are audit findings and failing an audit can carry significant consequences such as losing a certification or license to conduct business.

I'm hoping to bring both of these things to the WebSphere MQ world. The first tool I put out there is the "5 Minute Audit". This is a very simple checklist that just looks for the existence of any inbound channel with blank MCAUSER, SSLCIPH and SCYEXIT attributes. This combination means that the queue manager allows anonymous administrative access. Unfortunately, most MQ shops out there will fail this test. On the other hand, if this vulnerability is fixed, it addresses most of the security problems of a sloppy MQ implementation so giving this checklist to auditors can make a big difference. Fixing this issue goes a long way to paying off all that deferred risk that has been accumulating in your MQ network over the years.

The other thing I'd like to do is to create some publicly available MQ scanning tools. Or to be more precise, I'd like to assist *someone else* in creating those tools. My feeling is that such a tool would be better served through open source, or at least available source like BlockIP2, and broad community support. We've seen in the case of BlockIP2 that community contributions have resulted in code that is fairly robust, portable across a wide variety of platforms and is updated reasonably quickly when weaknesses are discovered.

But these things just tweak the system by making MQ security a measurable quantity. At the end of the day someone has to actually implement something. And that's where you come in. From an organizational perspective, making the investment in security requires a certain amount of faith. Chances are your auditor still doesn't know about WebSphere MQ and you are not likely to fail an audit any time soon. Or maybe you are not in an industry that is subject to regulation and audit. It may be that the only pressing reason to secure the MQ network is just because *it's the right thing to do*. As a WMQ administrator, look at your buddies across the aisle working on the web application servers. They don't even think of running the app server administrative console in plaintext. They only use



SSL. With scanning tools and standardized audit criteria, it won't be long before MQ will be administered the same way. Considering how little it costs compared to what's at risk, it doesn't take a lot of faith – just a willingness to do some basic authentication for administrative connections.

What would be really cool is if people became more aware of these risks so that companies could differentiate themselves by addressing them. For example, if you have a choice of two vendors who require MQ connections through your firewall and only one uses SSL, pick the one who uses SSL. Their balance sheet may not look quite as good as the next guy because they made that investment in security, but the day after the other guy is breached you will be thanking yourself. If you are that vendor, then make sure to make a lot of noise in the market about that additional level of security you offer. Use it to your competitive advantage.

If you are an audit firm or QSA and are qualified to assess WebSphere MQ networks, you can use this as a differentiator. Advertise it on your web site. Put it in the brochure. It may not be a big seller now but it will be and when it does you will have the first mover advantage. Or look at it from the reverse angle. Suppose you do an audit and don't look at the MQ network and then your client discovers that it's wide open. Seems like that would be a big credibility issue and might cause that client to go look for another auditor – maybe the one who advertises that they know something about MQ.

So to summarize, I think that the poor implementations in WebSphere MQ security represent an enormous risk. I believe that risk is asymmetric and at it's worst a breach could result in the collapse of companies who are exposed. But I also believe that this large gap represents an opportunity for companies to differentiate themselves and generate some revenue. Overall, MQ security is so poorly implemented that the market is quite large and companies getting in now will have first mover advantages. Currently auditors and assessors can move in this space but once the prevailing practices begin to shift, it will generate a need for skilled practitioners at all levels – developers, architects, administrators, and so forth. In short, this big gap and all this accumulated risk represents a big opportunity for those with foresight, vision and the willingness to take a reasonable risk. Not moving in this space? Well, that's like paying at making model rockets out of metal cartridges and gunpowder. You will be lucky if a finger is all you lose.

Let's take a quick break and then lighten things up with a little listener email.

Break.

Welcome back, I'm T.Rob and you are listening to The Deep Queue, a podcast of indeterminate depth. A listener sent an interesting email with the subject “WebSphere MQ as practised in the real world”. I was expecting an argument about how expensive or tedious MQ security is or how it isn't needed on the “trusted” internal network. I tend to get a lot of email like that. So I was pleasantly surprised to find it was a cartoon with the file name “Safe MQ at xxxxBank.jpg”. I'll post it on the T-Rob.net blog entry for this podcast but in the meantime I'll describe it. There's a house with “MQ” on the roof and a padlocked front door that has “SSL” written on it. Off to the left is a guy with a laptop and some padlocks hanging off of him saying “you should use this new crypto, it's much better.” On the other side of the house is a side door marked SVRCONN which is, of course, wide open and there's a



masked, gun-toting robber running in. I have to say I would have liked this cartoon if it stopped there but the guy on the left has T.Rob written across his chest and has a long drooping mustache.

This gave me a really good laugh when I got it. Made my day, actually. Still, I'm a bit of a perfectionist and there are a couple things I'd change if I could. First, there's a candy-cane tree in the front yard making this a bit fairy tail-ish. Unfortunately, it's more reality than fantasy so I'd lose the candy cane. Second. My character has a big dopey grin on his face like he's happy about the situation. No way I'd be that happy in this situation. In all fairness to the author though, I usually do have a big dopey grin on my face so maybe he was just being authentic. In any case, thanks to the author Mike Dent for giving me permission to use the cartoon.



I'm packing up tonight to fly out to IMPACT tomorrow so that's all for now. If you are at IMPACT, please stop by one of my sessions and say "hi". If you are on Twitter, the conference hashtag is #ibmimpact. You can also participate in the IMPACT social network at Event Vue at the URL <http://impact09.eventvue.com/events>. Thanks, and we'll see you next time on The Deep Queue.

Links for this episode:

IMPACT conference on Event Vue: <http://impact09.eventvue.com/events>

Twitter #ibmimpact hashtag: <http://twitter.com/#search?q=%23ibmimpact>

