

The Deep Queue

WebSphere MQ Security Podcast
Episode #11, May 25, 2009
Security breaches are not news?



Hello and welcome once again to The Deep Queue, the podcast that is 100% dedicated to WebSphere MQ security topics. I am your host

T.Rob and I am coming to you from Charlotte, North Carolina where it is Memorial Day weekend and around these parts, that means Speed Week. I live not too far from the speedway, which would be great if I were a NASCAR fan. But since I'm not, the phrase "Speed Week" to me translates roughly to "stay the hell away from Concord and the downtown Charlotte areas." I don't object to racing or race fans mind you, I just don't like crowds, traffic or beer – three things you apparently have to be extremely tolerant of to participate. And not necessarily in that order or in equal measure. For example, I think that if you just like beer and crowds they'll let you slide on liking traffic. For that matter, if you are REALLY into beer and are the type to try to smuggle a pony keg into the speedway in a baby buggy, liking traffic and crowds becomes optional.

No, I stay well clear of the racing crowd so instead this weekend I loaded, hauled, unloaded and spread 125 bags of mulch. I've never shied away from hard work, even at my heaviest weight, but I have to say this job was MUCH easier now that I'm relatively skinny. On previous podcasts I've mentioned having dropped a lot of weight. In May of 2007 I was 370 lbs. That's about 168 kg or 27 stone. These days I'm just under 200 lbs which is 90kg or about 14 stone.

Looking back, one of the interesting things was that occasionally I would meet someone who had no mental filters and their reaction would be along the lines of "My God, look at yourself! Why don't you do something?" It was easy for an outside observer to see there was a problem. I mean, here's a guy in the 98th percentile of human weight. But from the inside looking out, it was a different story. From my perspective it happened so gradually that I constantly reset my baseline for what was normal without realizing it.

I wrestled in high school and dropped weight to get into the 160 lb weight class. After that, I gained about 10 lbs per year for 20 years. During that whole time I never felt as though my weight was preventing me from doing anything I wanted to do. What I didn't realize is that I had been unconsciously adjusting the things I wanted to do so that they were always within my limitations. There was always a point in time, a line drawn in the sand, beyond which I swore it would be time to take some action and drop the weight. It's not like I didn't know I was big. But that line in the sand always seemed to move away with the horizon. I was never "too big."

But two years ago this weekend, I finally crossed that line in the sand and did something about it. I had the good fortune to find a physician who showed me, in a way I was able to understand, that I'd crossed the line a long time ago. I was no longer risking my health, I was risking life itself.

At this point in the show I always stop and wait for you to ask "Nice story and all T.Rob, but what in the wide world of sports has this got to do with WebSphere MQ security?" Go ahead. I'll wait.



<pause>

Well, I'm glad you asked. The state of security in the MQ messaging networks as deployed around the world is me, two years ago. We are well beyond the point of risking merely the health of the company. In today's world, exposing administrative access on the messaging network endangers the very life of the company. A data breach could result in corporate death. Now it's my turn as an outsider to say "My God, look at yourself! Why don't you do something?"

The difference is that instead of the gradual accumulation of weight, the enterprise network – your network, in fact – is increasingly at risk. The perimeter defense at the edge of the trusted intranet is crumbling and it's no longer good enough to install MQ with the default settings, or to give out administrative access to all users and applications. If you are a regular listener, you know I make this point every other podcast and you may feel like you've heard this all before. But remember, I give you different evidence each time. There's no shortage of breaches to talk about. In fact, *the fact that there is no shortage* is what I want to talk about today.

Perhaps you've heard about the data breach at UC Berkeley? The breach started on October 9th of last year and went undetected until April 9th, which is a period of 6 months. The Social Security numbers and medical histories of 160,000 current and former students and family members were stolen as a result of the breach.

The breach itself is tragic. When card numbers are stolen the bank can issue a new card. It's expensive but the damage can be measured and contained. In the Berkeley breach what was stolen were Social Security numbers. These are not immediately convertible to cash like card numbers are and the government doesn't reissue them. What they are more useful for is to meet the burden of proof to establish an identity for such things as opening a mortgage or home equity line of credit. It's possible that the people affected in this breach will not be victimized for years to come but that when they are, it will be big.

But what I really want to call your attention to is the coverage that this event received from the folks at the Security Squad podcast from May 15th. The topic of the news item in their podcast was not so much the breach itself but rather whether it rises to the level of being a big news story. I want to play a snippets from the podcast. What you are going to hear is Rob Westervelt and Neil Roiter discussing the Berkeley breach.

[Editor's note: the portion of the Security Squad podcast that was replayed on The Deep Queue is transcribed here from the audio. The attributions of who was speaking may be incorrect and the translation may not be verbatim.]

Rob: Is this worthy of a major breaking news story, 160,000 records exposed at this university? And it just struck me that maybe this really isn't that big of a story when, you know, you've got millions of records breached at TJX. We've done stories about possibly millions of records at Heartland Payment Systems breached, and here we are with another university breach. We've done stories on a number of university breaches as well. Neil, what's your take?

Neil: I don't believe it's a major story because as you mentioned there've been so many of these and it almost gets to the point where we're talking about the breach du jour. Is it important? Of course it's



important and it affects a lot of people but it sounds like it was fairly typical of most of the breaches we've seen. It's people coming in externally. The records were exposed for something like six months, which is again unfortunately, something fairly typical. These breaks are usually external and they usually go undetected until either internally, or more typically somebody on the outside notices something unusual happening and reports it to the folks who own the data. So it's important to report these things as they happen to continue to highlight the problem, but I don't see anything unusual or outstanding about this case.

So what is being debated here is whether data breaches have become so common that there is or should be a threshold of financial damage or the number of records stolen below which a routine breach does not qualify as a big breaking news story. To their credit, the consensus among the Security Squad guys seemed to be that this was indeed worthy of reporting. Personally, I'm stunned that we've reached the point where penetration of the firewall and perimeter defenses has become so routine that there would even **be** a debate about the newsworthiness of such an event. To me, the frequency of breaches *is* the story! How can we possibly reconcile the notion that breaches occur so frequently that exposure of 160,000 Social Security numbers could even be considered routine against the continued existence of the term "trusted intranet"? If breaches are routine, then the intranet cannot possibly be trusted.

But you might be asking "Just how frequently are we seeing data breaches?" Fortunately, the folks at PrivacyRights.org keep a list of all publicly reported data breaches or credit card information. Their list includes all manner of data breaches, including printouts tossed into the trash. But it's possible to go down the list and count all of the ones that were network breaches. Here's the stats:

January 5th - an employee of the Library of Congress stole employee data and passed it to a relative who used it to open accounts.

January 6th - CheckFree reported it's domains were redirected to a spoof web site in the Ukraine where hackers collected data from users.

January 7th - Geeks.com reported that it's eCommerce web site was hacked names, addresses, phone numbers, email addresses, credit card numbers, expiration dates and pins were compromised.

January 11th - University of Rochester reported that a student database was copied to an external IP address. Data included Social Security numbers.

January 14th - Occidental Petroleum reported that a former employee emailed himself a spreadsheet containing employee names, addresses, employee identification numbers, birth dates, starting dates, retirement dates and Social Security numbers.

January 19th - Forcht Bank disabled customer credit and debit cards after a retail merchant reported their computer system hacked. The merchant is unknown and the hack breach multiple banks and multiple debit and ATM networks.

January 20th - Kanawha-Charleston Health Department reported that people who received flu shots from the agency since October, are being warned that their personal information may have been stolen by a former department temporary worker. Information included their names, social security numbers, addresses and other personal information.

January 20th - Heartland Payment Systems: After being alerted by Visa and MasterCard of suspicious



activity surrounding processed card transactions, the company found evidence of malicious software that compromised card data that crossed Heartland's network. This incident may be the result of a global cyberfraud operation. More than 600 banking institutions affected, more than 100 million records stolen.

January 23rd – Monster.com database had been illegally accessed and user IDs, passwords, names, e-mail addresses, birth dates, gender, ethnicity, and in some cases, users' states of residence were stolen.

January 30th – Coos Bay Department of Human Services: A scammer made off with Social Security numbers after sending a virus online to a computer at the Department of Human Services office. A application that was installed recorded keystrokes and sent them to an external address. The information was taken from Coos County residents.

January 31st – Honey Baked Ham: A computer server stocked with credit-card information was stolen from a store. Customers might be at risk.

OK, that's just January, and the eleven incidents I read off are just the ones where the internal network was breached. I only counted incidents where a) the network was hacked from outside, or b) an insider stole the data off the “trusted intranet” or c) computers or data storage media were stolen from on site. Incidents I did not count that are on the list included accidental publication of data, discarded data and laptops where it was not clearly stated that they were stolen from the office. In addition to the 11 incidents in January, another 49 incidents met my criteria in the 3.5 months between February 1st and May 19th of 2009, which was the latest reported incident as of this podcast.

In the first 139 days of 2009, there were 126 incidents that made the list. Of those, 51 met my criteria for what constitutes a breach of the internal network. That's a little more than one every three days. And you might have gotten the wrong impression of who was breached, based on the names I read from January. Looking at 2009 to date, the victims included Symantec, Kaspersky Labs, Sprint, the US military, Los Alamos Nuclear Laboratories, a central bank, CheckFree, Lexis/Nexus, and of course Heartland Payment Systems.

Other things to consider are that the list I'm referring to is focused only on breaches that expose individuals to identity theft and only those breaches that occur in the United States. If for example there was a breach in which trade secrets or insider knowledge were stolen, it wouldn't be on this list. Whatever breaches occurred outside the United States during the same time period are not on this list.

Also, during the same time period in 2005, which is how far the list goes back, there were less than 50 incidents reported. That is close to a 300% increase over a four year period. Of course we don't really know how accurate the list is or how the data collection methods changed in the same four years. But I'd argue it doesn't matter. Assume the list is completely accurate, then we can conclude that breaches are increasing in frequency at an alarming rate. On the other hand, if we assume that the reporting and data collection is getting better and the breach rate has remained steady, then we can conclude that the situation is and has been far worse than anybody thought!

To me, **this is** the big breaking news story. The fact that breaches of the trusted internal network are now so common that we can debate, with a straight face, whether exposure of 160,000 Social Security numbers is newsworthy, is in itself news. Instead of looking at the size of the breach or the financial impact of the breach, why aren't we reporting on the frequency and nature of the breaches? Most companies take an attitude of “it won't happen here”. What's more likely to change that attitude, reporting that some other company was breached, or reporting that the rate of breaches is doubling



every few years?

From your perspective inside the company, the erosion of perimeter security probably seems gradual, if you notice it at all. There is probably a line in the sand that you've drawn which, when you cross it it will be time to do something about the security of your messaging network. And I'm guessing that each year that goes by, the line keeps moving toward the horizon. But just as people looked at me when I was 370 lbs and they couldn't understand the internal processes that prevented me from seeing an urgent problem, I as an outsider looking in don't see the history of your organization and all the things that were prioritized above security over the years. All I see is a probability approaching statistical certainty that your internal network will be breached and you cannot afford a messaging network that is completely exposed to such an incident. Having the benefit of that external perspective, it's my turn to say "My God, look at yourself! Why don't you do something?"

Whew, well that's a bit of a heavy topic. Why don't we take a break and then I want to talk a little about the CERT Security podcast and their discussion of "never events".

Break.

Hello again and welcome back to The Deep Queue WebSphere MQ Security Podcast. In the first segment we talked about how data breaches on the trusted intranet have become so commonplace that the Security Squad guys had serious discussion over whether the UC Berkeley breach was a newsworthy event. In this segment I'd like to introduce you to the concept of a "never event", something I first heard about on the May 5th episode of the CERT Security Podcast with Julia Allen and her guest Bob Charette.

On the CERT Security podcast, Bob Charette described what a "never event" means in a medical context. The term was coined by the National Quality Foundation to describe serious, preventable medical events. The NQF formally refers to these as SRE's or Serious Reportable Events. Informally, they are known as "never events" because they should never happen. The list includes things like performing a procedure on the wrong body part or wrong patient, administering the wrong blood type, or leaving a surgical instrument inside a patient.

In absolute numbers, the NQF reports that more people die from medical errors yearly than die from car accidents. Estimates range from 44,000 to 98,000 deaths per year are from medical errors. But as a proportion of all procedures, these are rare events. Given the number of procedures and hospital stays, even 100,000 events is statistically a very small proportion of the total. In order to drive that number down, the list of "never events" was created. The list contains specific events that can be both measured and reported. As of 2008, NQF reports that 25 states now require reporting of medical events and many more are considering legislation. In addition, government and private insurers are now including provisions which do not require them to pay for some preventable conditions related to "never events." This puts a tremendous incentive on the caregivers to implement rigorous procedures with accountability and transparency.

The reason this was discussed on the CERT Security podcast is that Bob is campaigning for software "never events". I'll play a little bit of that and let him explain:



Julia: why do you think this might be a useful concept particularly now?

Bob Charette: Well for a fairly number of years, and you're aware of this because we've had conversations about it, I've been trying advocate for something similar to medical "never events" and have some type of list in regard to software, either in terms of development or process, etc.

Unfortunately I haven't really been able to make much headway because it was pretty hard to get agreement from my colleagues in either academia or in industry as to what we could really call a software "never event". First of all part of the reason is one of language and terminology. In the medical field, "never events" are very, very extreme events. They're things that really don't happen at all in any type of major statistical quantity. Whereas in software we're constantly surrounded, it seems, by problems, so a lot of my friends said we probably should call "In hopes of a software never event" or maybe better "Software ever events". In fact my friend Martin Thomas, who cofounded the UK software company Praxis which is a high reliability software company and he's a visiting professor at Oxford University Computing Laboratory, wrote me when I was talking to him about this that he's a strong supporter of having some type of initiative where we could eliminate things that shouldn't occur. But as far as he could recall, the software industry has never learned from a mistake and created a "never event." So part of it has been some push back in terms of how do we actually describe these things.

But recently it's come to my attention that a coalition of more than 30 US and international cyber security organizations led by MITRE and the SANS Institute and CERT jointly released a consensus list of what they're calling the 25 most dangerous programming errors that lead to security bugs and that are used to allow cyber crime and cyber espionage and cyber warfare. And they're calling this the Top 25 Errors Initiative.

And what they say, and I'm going to quote from their press release, is that "the impact of these errors is far reaching. Just two of them led to more than 1.5 million web site security breaches during 2008 and those breaches cascaded onto the computers of people who visited those web sites, turning their computers into zombies." They also said that "most of these errors, most of the 25, are not well understood by programmers. Their avoidance is not widely taught by computer science programs. And their presence is frequently not tested by organizations in developing software for sale."

So when I looked at that list I felt that well maybe this would be a good time to revisit the idea of "never events" and to see whether or not, given the increased frequency and severity of IT security incidents, that maybe this would be a good time to start a new conversation. The idea seemed pretty obvious in retrospect, especially if there's a list of agreed upon errors and ways to prevent them.

Well, I rather like the idea of a software "never event". As I said earlier, these events are standardized, measurable and reportable. Recalling the discussion in the first segment, this level of discipline is lacking in our current system. Currently the reporting part is left to the media and they are not sure whether a breach of 160,000 Social Security numbers merits reporting. But in the context of a "never event", there is an objective criteria for whether something is reportable and the rate of events, whether that rate rises or falls over time, is newsworthy in addition to the events themselves.

As Bob mentioned, the Top 25 Errors Initiative is the closest thing yet to a software "never event" list.



It at least enumerates those definable, measurable, reportable events that are responsible for the majority of breaches in terms of numbers and of impact. The main difference between this list and the NQF list is that the NQF list is centrally reported and tracked.

I thought it would be an interesting thought exercise to imagine what a WebSphere MQ “never event” list might contain. Here is what I came up with:

1. Anonymous administrative access should never be exposed. It's easy to lock down or restrict channels. We should not be giving administrative access to the entire intranet.
2. Sensitive data should never be sniffed off of a WebSphere MQ channel. The channels have been capable of running SSL for several years now. If card numbers or PII are running over them, the channels should be encrypted.
3. Administrative access should never be granted to legitimate ordinary users and applications. The security MQ model allows for differentiation of administrators and others. We should not routinely be putting everyone in the mqm group.

I only have three listed but these three account for most of the exposure that I see in the field right now. Let's take a closer look at them.

First and foremost of course is not to expose administrative access to anonymous users. In the messaging world there are many, many different business requirements for security and there are not many pre-defined security models out there one can just pick up and apply. There is one however, and that is to lock down administrative access to the queue manager. This one is at least well documented, measurable and mature.

Going back to the weight-loss analogy from the first segment, this is the equivalent of “eat less, exercise more.” It is the advice that everyone knows they should follow but almost nobody actually does. Chances are there's a line drawn in the sand somewhere that once you cross it, it will be time to address administrative access on the MQ network. But that line keeps moving toward the horizon. There is always something higher priority in front of it, just like for me my job, my family, my hobbies prevented me from exercising more.

But in terms of MQ security, there is no more basic or fundamental task than to prevent anonymous users from obtaining administrative access. Once that happens, game over, you might as well give up. Administrative access means the ability to read, write or alter any message, the ability to remotely execute arbitrary code on the MQ server and the ability to steal the SSL keys. I don't know of any company that routinely grants access to sensitive databases to everyone on their intranet but almost all companies grant anonymous access to the messaging system which accesses that same database. This should never happen. It's easy to prevent and the techniques are well known. My tutorials on this are published on <http://t-rob.net> and on IBM's developerWorks technical journal. Others have also written on the subject.

Next on my list is that the ability to sniff card or other sensitive data over an MQ channel should never happen. Given that the MQ channels can use SSL, why on Earth would it be necessary to send



sensitive data in clear text? Now, I've heard objections that it's too hard. But we are talking about the continued existence of your enterprise. Writing an application with 100,000 lines of code is hard. Getting a requisition to hire a new employee is hard. Generating and exchanging a couple of self-signed certs is easy by comparison. I know of plenty of MQ shops that are running SSL successfully.

If you are not running SSL and want to be, dive in and do it. If you run into problems, ask for help on the list server or on <http://MQSeries.net>, or open a PMR and get IBM support to help you out. Especially open a PMR if what is holding you back is that it's too hard to do. IBM needs the feedback from you if the product is ever going to improve. So if you think it's a pain in the butt to have to delete all of the default Certificate Authorities every time you create a new SSL keyring, let IBM know about it. Just don't let it stop you from completing the implementation.

My final item was that administrative access should not be granted to ordinary users or applications. This again is very simple to fix by setting an MCAUSER on the channel used by that user or application to connect. If it's a local application, just don't put it in the mqm group. Create an application group for it and authorize that group for just the access it requires and nothing more. The practice of allowing applications and users to have administrative rights because it's easier than properly authorizing them simply has to stop.

Go back and look over that breach list I talked about in the first segment, and see how many of those breaches were employees and others with legitimate access. There were also a number of them that were by someone with internal access getting a virus or not having proper physical security. When a legitimate user or application is over-authorized you run the risk of a malicious internal employee stealing data or a hacker that compromises that employee's system and uses their elevated access to steal your data. If the user or application is properly authorized, the exposure is only the data that user is legitimately authorized to. Although that might be bad enough, it is certainly worse to allow that user access to everything on the messaging network.

So that's my list of three “never events” for WebSphere MQ. Restrict anonymous administrative access, encrypt sensitive data on the wire and don't over-authorize legitimate users. Coincidentally, all three of these items are present in both the PCI Data Security Standard and the recently published Top 25 Programming Errors list. They should be, by any reasonable standard, on anyone's WebSphere MQ “never event” list. Like a 400 lb human, the risk we are talking about here is not merely the health of the company, it is potentially the life or death of the company. Addressing these issues is like exercising more – there are a million other things that seem more important and provide immediate results. And like weight loss, some precipitating event is usually required before action is finally taken. For some people it can be a heart attack or onset of diabetes. For me it was a caring physician who explained my risk in a way that I was able to understand. It is my sincere hope that through these podcasts I can do for you what my doctor did for me and explain the risks of an unsecured MQ network in a way that is understandable and inspires you to take action. If there is anything I can do to help you take that next step, please contact me.

That wraps up this episode of The Deep Queue for May of 2009. This is T.Rob, signing off saying, “My God, look at yourself! Why don't you do something?”



Links for this episode:

University of California Berkeley Data Breach

<http://datatheft.berkeley.edu/news.shtml>

Security Squad, SearchSecurity.com podcast for May 15, 2009

<http://itknowledgeexchange.techtarget.com/security-wire-weekly/squad-data-breach-burn-out/>

PrivacyRights.org Chronology of Data Breaches

<http://www.privacyrights.org/ar/ChronDataBreaches.htm#CP>

BankInfoSecurity.com – List of banks reported to have been affected by the Heartland breach tops 600

http://www.bankinfosecurity.com/articles.php?art_id=1200

National Quality Forum – Serious Reportable Events (a.k.a. “Never Events”)

<http://www.qualityforum.org/projects/completed/sre/fact-sheet.asp>

CERT Security podcast series for May 5, 2009

<http://www.cert.org/podcast/>

WebSphere MQ Security Heats Up – Blog post with downloadable setmqaut scripts to secure administrative access to WebSphere MQ.

<http://t-rob.net/2008/07/08/websphere-mq-security-heats-up/>

