

Jack and the Magic Security Beans

Posted At : November 16, 2005 1:51 PM | Posted By : Mark Kruger

Related Categories: Security

Customers often show up on our doorstep with significant issues looking for a solution. That's a good thing. We like to think we can provide them. But so often there is a perception that there *is one solution* that is going to fix a particular problem. That is often not the case. The truth is that fixing a sick server or fine tuning an application or database may require many steps. When it comes to security this is especially true. Customers will often try to grasp security by fixating on 1 approach or 1 item that seems the most crucial to them. But there are no magic beans when it comes to securing your application. Just defining what the customer means by *secure* may be your biggest challenge.

What does "secure" mean

Does secure mean that a hacker can't "bring down" your site? Does it mean he cannot gain access to a members area or customer login area? Does it mean that a malicious person can't destroy data? Or does it mean that he can't "read" data you don't want him to read? See what I mean? There are definitely lots of nuances to security. You can think of security in terms of layers. In fact, that is the language often used by security experts - "layered security" or "defense in depth" to borrow a military term. Think of a skirmish line followed by heavier troops backed up by tanks and you get the idea.

Network Security - Things like firewalls, SSL and VPNs all work at the network level. The purpose of such devices and protocols is to keep your data from being "sniffed" out - to keep anyone from "seeing" the content of your network packets. Firewalls and protocols keep out bad guys who use blunt tools to figure things out about your network and infrastructure. While this may *not* be the most important part of your security strategy you should *never be without it*. It's kind of like the clothes. No one comments when you have them, but you definitely draw attention without them.

Operating System - Your next line of defense is your operating system. Yes it is possible to configure a secure Win2k or Win3k environment - even as a web server (when you Linux guys are finished snickering I'll continue.... thank you). The operating system of a server should be "stripped" of any but the essential services. That means disabling all those extra services and helpful options that come with the default installation of a server. Even in the Linux world, if you install Redhat in the default installation mode you have a lot of work to do to lock it down. Create password policies, change the default administrators account, work with your ACL's etc. The best tip I can give you is to have a strategy that installs a new server with a bare minimum of stuff and services on it, then build it up from there. Don't try to do the reverse of installing everything and then "locking it down".

Application Security - ok, so you have an air tight firewall and a bastion host for a server. You are secure, right? In fact, "bad code" is responsible for most breaches of security. No amount of security at the network or OS is going to keep people out if you allow 3 letter passwords for example. Or (as in one recent discovery) all people to use the word "password" - still the most common password on the web (sigh). Pay attention to security as you write your code. Validate form inputs. Use data binding. Implement password policies. Use the application level techniques and tools at your disposal to lock down your code.

Data Security - You can have excellent and inscrutable code, but if you store everything in an access file right off the root of your website all is lost. Don't laugh, a couple of years ago I convinced someone of their need for security by trying a few URLs like "http://someDomain.com/database/someDomain.mdb". A couple of tries while I was chattering away and I suddenly had all his customer data in front of me. A neat trick if you can pull it off. Whether you are using Oracle, DB2, MS SQL, MySQL or carrier pigeon you need to keep security in mind as you build the framework of your site. Where will the usernames/passwords be stored? What permissions will be given to the user? How often will the data be backed up? What about transaction logging? These are all issues that need attention.

Backup - what? This is supposed to be about security right? News flash *backup is a part of security*. You have to have a way of recovering if (and when) a malicious user hits you with a broad side and manages to get a shell into the boiler room (sorry, I've been reading civil war history lately).

So, as you can see, there are *many* issues involved with solving a single problem like security and no one solution is going to solve it for you. You have to consider all the levels at which the data or code can be accessed and account for them. I haven't even spoken about physical security. Is your airtight code being hosted in the basement of a guy living with his mother and her hell's angel boyfriend for example - it could happen! The point is to be broad in your approach. When it comes to security - sweat the details. Just one incident where you save the bacon will more than pay for all of your dire warnings.