FILEMAKER CERTIFICATION TOPIC #6: Securing Your Databases (Part 1 - Overview)

This paper summarizes the security picture in FileMaker 7, to clarify how the product can be viewed in this area. It does not get into details about how to actually use the various features included in the product.

Primary Resources

- **FileMaker's "Security" Web Page**
  http://www.filemaker.com/support/security.html
- ***FileMaker 7 Security*** [white paper]
  http://www.filemaker.com/downloads/pdf/whitepaper_fm7_security.pdf
- ***Upgrading to FileMaker 7: How to employ the new, advanced security system*** [tech brief]
  http://www.filemaker.com/downloads/pdf/techbrief_security.pdf

Customer's Security Concerns

	Customer Category	**What They're Concerned About**
1	Application Developers	data integrity, intellectual property, maintainability
2	System Administrators/IT	leveraging existing infrastructure, protecting network resources, securing organizational data
3	End Users	accountability, reliability/availability, ease-of-use

Types of Attackers

	Characteristics of Attacker	**Approach to Thwarting Attackers**
1	Remote outsiders, possibly with some internal knowledge (e.g., ex-employees)	built into FM, via authentication schemes, etc.
2	Local insiders with network access	
3	Local insiders with admin access to computers hosting FileMaker applications/data	FM insufficient by itself to stop these; requires consideration of other security tools (firewalls, disk encryption, etc.)
4	System crackers with reverse-engineering skills and insider network access	

What to Tell IT About FileMaker 7

First of all, people need to know that FileMaker 7 is a complete revision of the product. One of the explicit goals of this revision was to address the concerns of IT professionals who, with good reason, saw previous versions of FileMaker as second-class IT citizens (at best), not worthy to be given sanction or server-room support. But FileMaker 7 is not your father's FileMaker. Security-wise, it's a completely different product.

	Area	**Feature in FM 7**	**Why That's Good**
1	*File Storage, Communication, Data Visibility*	FileMaker files are stored as Unicode, not ASCII text, and local cache is encrypted with fast proprietary algorithm	Casual users can't just bring up files in a text editor and see data values, hackers can't glean useful information by getting the cache
2		All FileMaker network communication is performed with CORBA formats; server-client traffic, incl. server administration, can be completely encrypted (SSL secured using standard, PEM-format x.509 certs.)	Hackers can't get anything from tapping the line, and can't hijack the session; certificate scheme (by default, self-signed) can be enhanced with 3rd-party technology if desired
3		Web Publishing uses standard web servers (Apache or IIS), can restrict access by IP address, and can secure channel using SSL	Web service much more reliable and maintainable, and can be secured to protect FileMaker-based data; uses standard ports
4		Server, web access can limit what meta-data users can see in a way not possible earlier (e.g., available databases screen in Open Host..., XSLT code can prevent argument-change hacking via web, etc.)	Reduce potential for learning about design of system without having legitimate access

5	*Authentication*	FileMaker now uses accounts and passwords, not just passwords, for all access to databases	Accountability becomes possible, each user having a distinct footprint; server/client, ODBC/JDBC, web access all use same system
6		Passwords are now verified on the serving host (account info no longer sent down to requesting client)	Authentication information stays in well-protected environment
7		Passwords are stored as salted one-way hash values (PKCS #5 PBKDF2 algorithm) in the database, and sent across the wire obscured by proprietary modification to standard cryptographic algorithm	Passwords are not stored in any accessible format, and the only authentication information crossing the wire is a well-protected password key
8	*Administration*	FileMaker access privileges, including home grown privileges, can be tied to accounts	Access can be controlled architecturally, not only via interface scripting
9		FileMaker's approach to accounts/passwords/privileges is now logical	It's actually possible to create a rational access/authentication system without learning a bizarre proprietary approach
10		Much of the account administration activities (creation/deletion/password change) can be scripted	Allows automated account management strategies
11		All FileMaker tables for a given solution can be kept in a single file, whose access setup applies to all the tables in the file	Creates easily maintainable, uniform security management
12		All login attempts are logged, in standard text-based log file, and web access can be logged in detail	Access can be audited, and analyzed using standard log analysis tools
13		Server backup approach revised to reduce impact on current users; database schema changes and administrative activities can be undertaken remotely without requiring system to be taken down or restarted	Much less down time; greatly enhanced system stability; greatly reduced need for non-IT folks to access to physical machine; ability to provide non-db-developers with administrative tools

Developer Best Practices — Some Suggestions

	What to Do	Why
1	Familiarize yourself with the current security methods, by reading documents and practicing the techniques	You can't use it properly if you don't understand it; it's especially dangerous if you are used to the old version
2	Always physically secure the machine serving the databases — lock the room, control account access, implement firewall, port-protection if appropriate	The database is much harder to hack if the hacker does not have physical access to the files; close as many doors as possible to access — and to attacks on the service too
3	Always create accounts/passwords for your databases	Establish access control and lay the groundwork for accountability
4	Even if you decide to have a blank password, give it the minimum access privileges required for that purpose	Don't trust to scripted interface to control access: build it in
5	If you convert from earlier versions, be sure to change account names and/or passwords so they no longer match, and review the entire account/password/privilege scheme	Too easy to expose the password to prying eyes; in some cases, privileges could be missing due to how conversion works; in any case, start living in the new world
6	If possible, serve the databases via FileMaker Server, and enable encryption of the data stream; if not possible, try SSH tunneling, VPN, Citrix, etc. to assure secure access	Protect your systems over the network, even in firewall-protected environments (client-client serving can't encrypt the stream like Server can)
7	Be very careful which scripts (if any) you give "Full Master Access" to	Be clear about what functionality you're putting in the hands of users
8	Don't presume you can control all access via scripting	Because you can't
9	If publishing via web, familiarize yourself with Apache or IIS security techniques, esp. SSL certificates	You can secure the web stream, but you have to understand how these servers do it — it's not just a switch
10	If publishing via web using XML/XSLT, institute code-based anti-hacking techniques	Prevent hackers from substituting arguments and getting to important data (e.g., metadata)
11	Consider using external authentication, if appropriate	Can simplify life (and may be politically expedient)
12	Backup regularly, and consider the security of the backup	Avoid data loss; don't make backup copy the weak link