FileMaker Workplace Innovation Platform Security Building Blocks

By:

Steven H. Blackwell
Platinum Member Emeritus, FileMaker Business Alliance

FileMaker Business Alliance Platinum Members are independent entities without authority to bind FileMaker, Inc. and FileMaker, Inc. is not responsible or liable for their actions.

The views and recommendations expressed in this White Paper are solely those of the author and may not necessarily reflect those of FileMaker, Inc.

FileMaker, FileMaker Go and the file folder logo are registered trademarks of FileMaker, Inc. in the U.S. and other countries. FileMaker WebDirect and FileMaker Cloud are trademarks of FileMaker, Inc.

© Copyright Steven H. Blackwell, 2019.
All rights reserved under both International and Pan-American Conventions.
Permission granted to users of FileMaker Workplace Innovation Platform products to distribute within their own organizations.

Version 1.1 February 2019

CONTENTS

FILEMAKER WORKPLACE INNOVATION PLATFORM SECURITY BUILDING BLOCKS

By:

Steven H. Blackwell
Platinum Member Emeritus, FileMaker Business Alliance

Version 1.1

The FileMaker Workplace Innovation Platform has evolved over the past fifteen years since the release in 2004 of FileMaker® Pro 7 to encompass now a robust suite of security features to protect the *Confidentiality, Integrity, Availability,* and *Resilience* of organizational data. This same set of Security Building Blocks also helps to provide protection to organizations from a wide variety of risks including business continuity, regulatory sanctions, criminal and civil liability, and reputational damage.

Even many experienced developers, however, do not have wide-spread knowledge of these Building Blocks, of how they work together with one another symbiotically, and the purpose they serve. Thus, the goals of this White Paper are intended to help address this issue:

- What are the Building Blocks?
- How do they work together?
- What purpose do they serve?

Answering these questions will hopefully enhance understanding of the Security Building Blocks among the developer community and within the end-user community as well.

◆The Authentication Process.

Effective security processes for FileMaker databases start with *Authentication*. We want to limit access to only those who are supposed to have such access and at the level of privileges that they should enjoy. We want to block unauthorized users; and, we want to prevent authorized users from escalating their Privilege levels.

This process starts with an ***Identity Assertion*** by persons seeking access. It continues with a ***Validation*** of that Identity Assertion. In short: *"Who are you? And are you actually who you said you are?"*

The FileMaker Platform relies on the use of credentials consisting of an *Account Name* and a corresponding *Account Password*. The presumption is that if someone who is seeking access to the database can supply a valid Account Name and the correct corresponding Account Password, that the person is a valid user and is who he or she claims to be. This is why passwords should be safeguarded and is also why each user should have an individual, unique Account Name and matching Account Password.

With depressing regularity, we see posts on various FileMaker-focused Lists and Forums either inquiring about or advocating the creation of alternative approaches to this core process of Assertion and Validation. I have called these ***Ersatz Systems,*** because they might give the ***appearance*** of Identity and Access Management, but they are susceptible to all manner of manipulation and subversion.

The FileMaker Platform can recognize and employ credentials from three different sources for access to hosted files:

- **Internal** to the database file
- Legacy **External**, meaning Active Directory, Open Directory, or Local Security groups on the FileMaker Server
- Three **OAuth Identity Providers**[1], namely Google, Amazon, and Microsoft Azure AD.

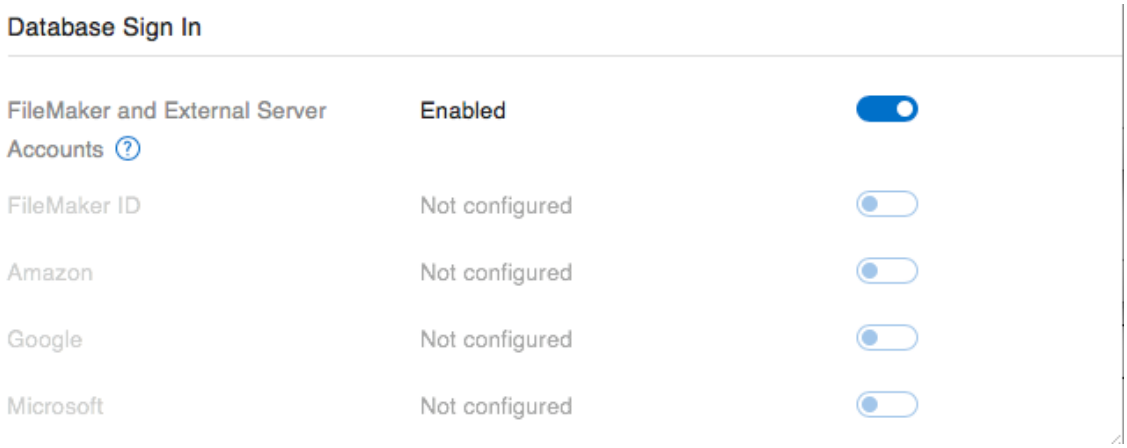

Figure 1. *Configure Options on FileMaker Server*

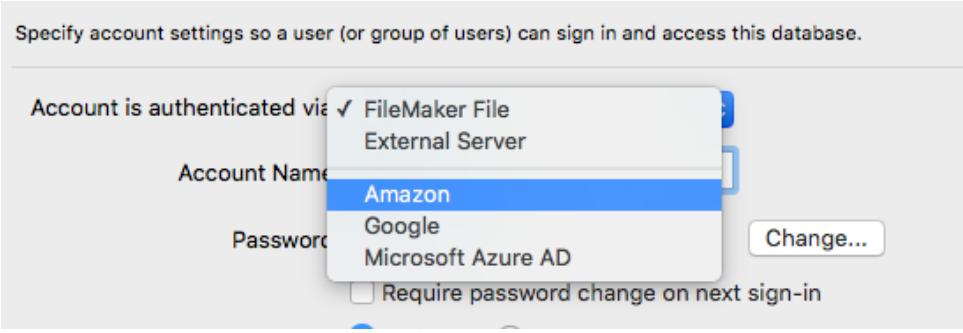

Figure 2. *Pick Amazon As An Example*

[1] I jointly authored a White Paper with the highly eminent developer Wim Decorte on this topic. https://fmforums.com/files/file/91-oauth-identity-providers/

Also see Wim's video on configuration of OAuth. https://www.youtube.com/watch?v=99BuLS3mCSY

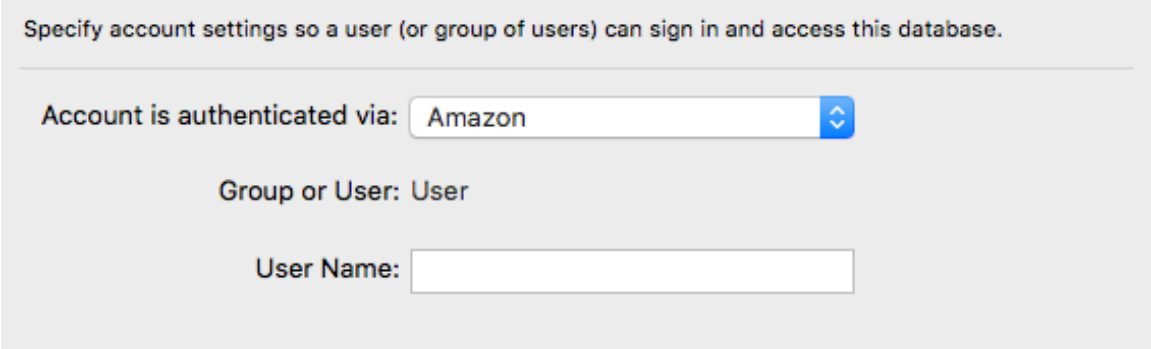

Figure 3. *The Result*

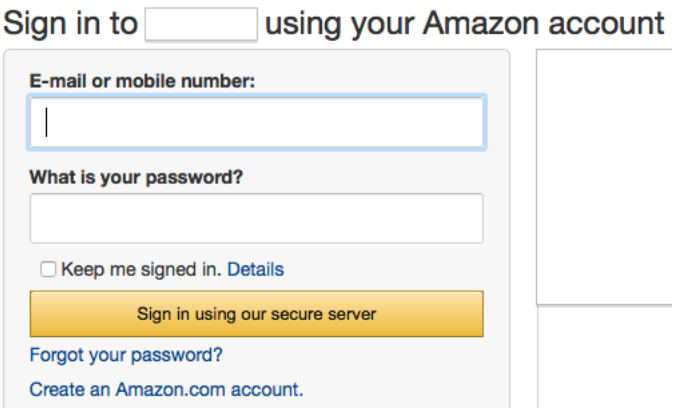

Figure 4. *The UI Presented When Seeking Access*

In the future, the advent of the **Zero Trust Model of Security**[2] together with ***Federated Identity Management*** will provide additional avenues of Identity and Access Management. Passwords alone are coming to be seen as insufficient validations of Identity Assertions. Passwords get lost, stolen, forgotten, and (even worse) shared amongst users. Thus, we are coming sooner rather than later to the threshold of requiring strong multi-factor authentication (MFA) most likely employing hardware tokens. The FileMaker Platform currently can leverage Active Directory and Azure AD for MFA.

But at base the core concept is this: *Who Are You, and Are You Who You Say You Are?* And if the Identity Assertion is validated, the user gains access to the database with a specific and defined set of privileges.

◆ Role-Based Privileges.

Once admitted to the FileMaker database, the user's actions and activities need to be governed by a *set of defined privileges* commensurate with his or her role in the

[2] https://fmforums.com/blogs/entry/2047-federated-identity-management-zero-trust-and-the-filemaker-platform/

business processes the database manages. The FileMaker Platform is especially adept at managing such privileges even to a very *discrete* and *granular* level. These privileges center on three main areas:

- What can the user access? [One table, but not another, for example].
- What work can the user do with respect to the data housed in the FileMaker database? [Create and edit, but not delete, for example].
- What Business Processes that the database system manages can the user invoke? [Run one report, but not another, for example].

◆Granular Access and Action Control.

Once admitted to the FileMaker database, the user's actions and activities are governed by the Privileges associated with his or her Account. This is one of the core reasons why developers need to give considerable thought to construction of these Privileges. A central concept in Information Security is the ***Rule of Least Privileges***. That concept states that a user should be given all the Privileges needed to accomplish the user's assigned work tasks, ***but no more Privileges***. We need to work in FileMaker database systems to prevent what is called ***Escalation of Privileges.*** That means that we need to prevent any user from obtaining additional and higher-level Privileges to perform tasks or to access information not otherwise allowed.

Granular Privilege control focuses on a number of familiar elements in the FileMaker Pro file:

- Data tables
- Certain aspects of individual fields within data tables, such as viewing or modifying them
- Record creation, modification, deletion, and viewing within a table
- Management of Global fields[3]
- Scripts, including accessibility, execution, and modification
- Layouts, including accessibility, modification, creation, and the behavior of instances of data fields on a specific layout
- Value Lists, including their accessibility and modification

If a developer allows end users to create new layouts—something that I do not really recommend as a general good practice—the developer can control the behavior of elements added to that new layout. Select the *Any New Layout* option in the Privilege Set definition and then select the desired level of control:

[3] Global fields are a special case since their behavior is **not governed** for the most part by the normal editing restrictions.

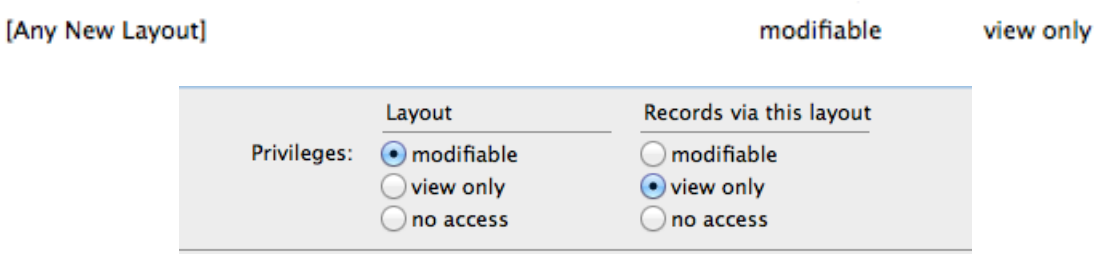

Figure 5. *Layout Control Options*

FileMaker Pro has the option for any Script to *Run Script With Full Access Privileges.* This action confers added power to the **Script**, ***not*** to the **User**, to permit actions not otherwise available to the user *via* the Privilege Set constraints. For example, if a user's Privileges prohibit the deleting of a record, the script could allow that action to occur under controlled circumstances that the script defines. This facilitates Business Process Management while still narrowing the user's permitted actions.

Here is an important caveat related to a number of Privileges found in the Privilege Set definition. In the section labeled *Other Privileges* are the controls for Printing and Exporting. **These controls apply only to the file where they reside.** This means that data can be printed or exported by a user from a different file if the table where those data reside is accessible in the other file. This is why the File Access Protection feature was introduced in FileMaker® Pro 11. There will be more on that later in this White Paper.

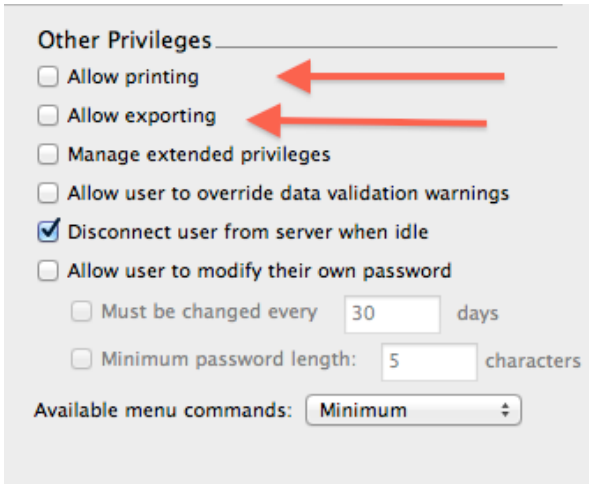

Figure 6. *"Other" Privileges*

◆ API Control.

Application Programming Interfaces (APIs) offer a way to interact with core FileMaker Platform elements including data, Scripts, and Layouts. They can also retrieve metadata about database files including such elements as names and ID's of various objects (*e.g.* Fields, Tables, Layouts, and, Scripts). These API's can also in some instances invoke Business Processes in the database file by causing Scripts to execute.

Sometimes this is good and desirable. These API functionalities extend the reach and functionalities of FileMaker Platform systems. In other instances, however, they can cause unexpected items to occur and provide avenues into parts of the file that the developer never intended to be susceptible to such access.

Three such APIs have been of particular concern. In some earlier versions of the Platform they rested in the file unfettered and totally accessible to invocation. These three APIs are ***AppleEvents*** (Macintosh OS), ***ActiveX Controls*** (Windows OS), and ***FMPURL Perform Script*** (both platforms). Now, however, since FileMaker® Pro 16, these APIs are all ***disabled by default.*** Developers must ***specifically invoke*** them on a Privilege Set by Privilege Set basis in order for them to work. This includes the default [Full Access] Privilege Set.

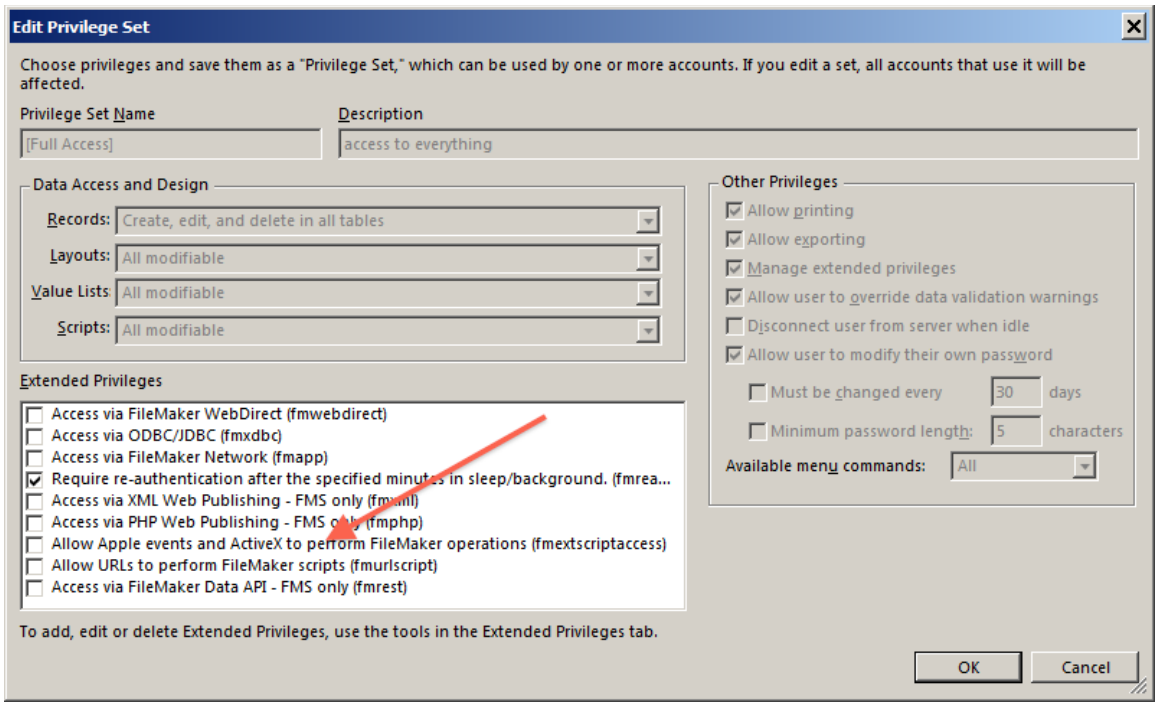

Figure 7. *API Privileges Must Be Explicitly Invoked*

The AppleEvents API was particularly powerful and susceptible to misuse by persons attacking the file. It could perform a number of actions:

- Return the names of all Layouts, Scripts, Tables, and Fields in the file not marked as <<No Access>> to the Privilege Set associated with the Account that opened the file.
- Navigate to any of these Layouts, irrespective of whether the Layout's name appeared in the Layouts Menu or not.
- Invoke and run any of the Scripts not marked as <<No Access>> to the Privilege Set associated with the Account that opened the file.
- Insert into or extract data from any fields not properly protected in the Privilege Set.

The ActiveX Controls and the FMPURL Perform Script APIs could invoke Scripts as well. And all these actions would occur outside the file and all of them could be activated by any user, authorized or unauthorized. This was a primary attack vector against both hosted and stand-alone FileMaker Pro files.[4] Additionally it was a principal method, although not the only method, to subvert and defeat the Ersatz Security and Log-On Systems as demonstrated by the well-known and highly-respected FileMaker developer Joshua Ormond several years ago.[5]

Thus, it is exceptionally good that FileMaker, Inc. has given developers to ability at a very specific and granular level to control the behavior of these three APIs.

There is, however, another API that can also access a FileMaker Pro file and perform a wide variety of actions on it. That API is **FileMaker Pro *itself*.** Through the construction of external data sources, one FileMaker Pro file can perform actions on the data and on the business processes found in a separate, different file. This is a rather common occurrence, especially in multi-file solutions.

Problems start however when the external file performing these operations is not a file created by the developer and is very likely unknown to that developer. These rogue, external files can perform a variety of actions on a file that is included as part of a FileMaker solution:

- Perform Scripts
- Export data
- Traverse Layouts including those thought not to be able to be seen
- View the contents of Scripts (in some instances) either directly or by printing them
- Print data
- In short, manipulate the data, User Interface, and Business Logic of the solution file.

---

[4] https://fmforums.com/blogs/entry/1535-the-filemaker-platform-api%E2%80%99s-are-your-friends-right/

https://fmforums.com/blogs/entry/1652-security-vulnerabilities-of-filemaker-platform-api%E2%80%99s-an-update/

https://fmforums.com/blogs/entry/1738-behavior-change-api-privileges-in-version-16/

[5] https://fmforums.com/blogs/entry/1512-a-conversation-about-2-factor-authentication/

In FileMaker® Pro 11, as previously noted, FileMaker, Inc. introduced File Access Protection[6] as a way to manage this process and to restrict unauthorized rogue files accessibility to authorized solution files. In the *Manage Security* UI of a FileMaker file, select the *File Access* tab. This will reveal a window where developers can require that any external FileMaker Pro file must have knowledge of a [Full Access] Account in the current file in order to access it. Likewise, this is where specific external files of the developer's choosing can received such authorization.

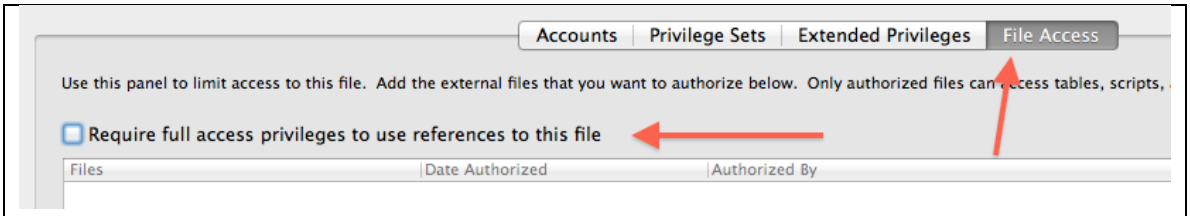

Figure 8. *Requiring and Setting Access Protection.*

At the present time, this option is not in force by default. Thus, developers ***must explicitly and affirmatively invoke it.*** When invoked, this option renders the rogue external file powerless to perform the actions previously described.

New in recent versions of FileMaker Pro and FileMaker Server is the ***Data API (fmrest).*** This functionality allows the FileMaker Platform to send and receive RESTful aware data and activities. Similar to the other APIs, developers must explicitly authorize this API to work on FileMaker Pro files. Developers must exercise particular care with Accounts, Passwords, and Privileges associated with this activity so as not to compromise credentials.

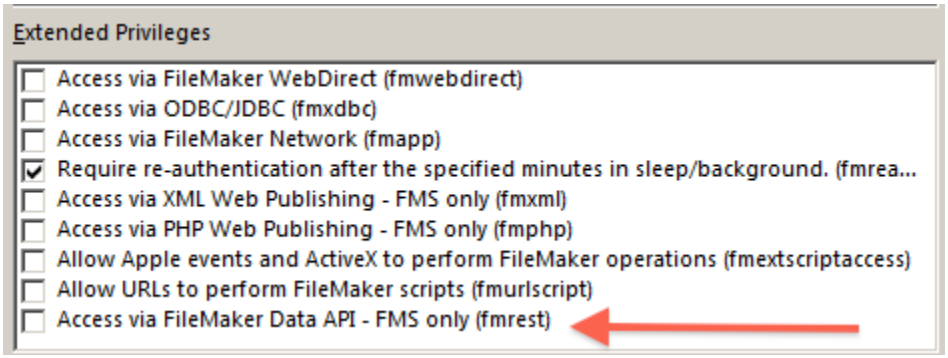

Figure 9. *Enabling the Data API (fmrest)*

◆ **Encryption.**

[6] http://www.fmpug.com/resources/security_schema_changes_filemaker_11

In addition to API Control and File Access Protection, Encryption is one of the most important FileMaker Security Building Blocks. Possibly Encryption is also one of the least well-understood and more frequently-misused Building Blocks. There are three type of Encryption in the FileMaker Platform:

- **Encryption Of Data In Transit.** As the name implies, this encrypts data traffic between FileMaker Server and FileMaker Pro.
- **Encryption Of Data At Rest (EAR).** This feature encrypts the physical binary file itself.
- **Field Level Encryption.** This feature encrypts individual fields in a table.

Encryption In Transit is invoked on FileMaker Server and requires that there be a valid SSL certificate installed on the server. In a White Paper[7] I jointly authored with the highly eminent developer Wim Decorte, "The Developer's Developer", we discuss this process as it applies to the latest version of FileMaker Server.

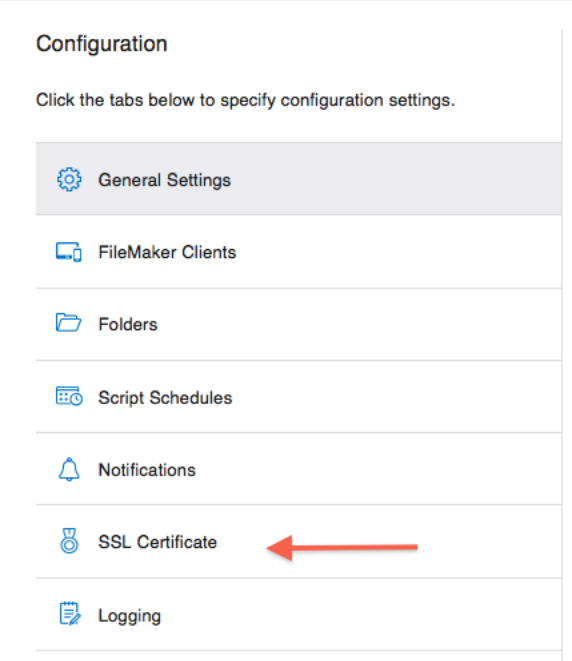

Figure 10. *Encryption In Transit on FileMaker Server*

Encryption At Rest options are in the Developer Utilities Tool section of FileMaker Pro Advanced itself. Developers employ this option by providing what FileMaker, Inc. calls an "Encryption Password." A meter measures and reports the strength of that special password; always use the *Strong* option. Such encryption protects the physical file, and it is also one of the best protections against the use of so-called "password crackers" that break into files.

[7] https://fmforums.com/files/file/102-filemaker%C2%AE-server-17-and-ssl-certificates-configuration-and-use/

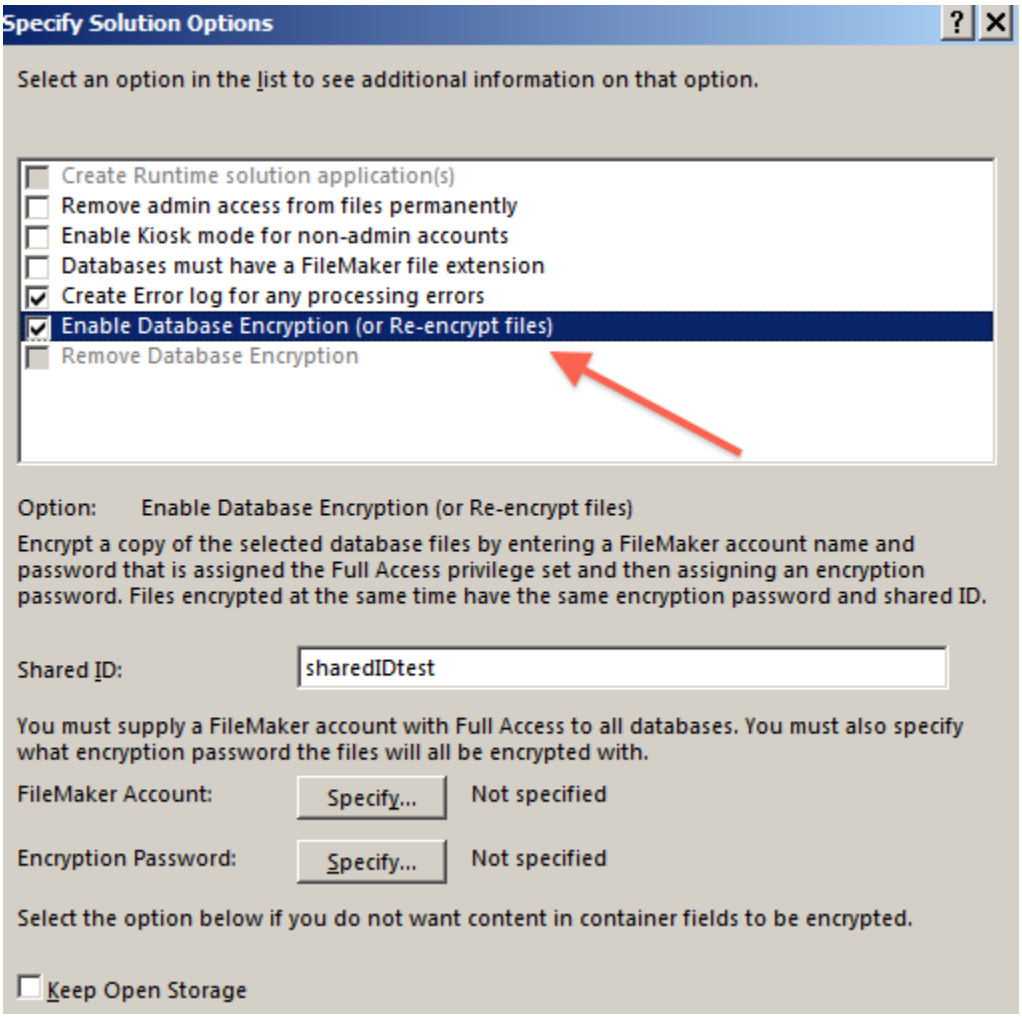

Figure 11. *Select EAR Option From Within Developer Tool*

Figure 12. *Use Strong Encryption Password*

Finally, *Field Level Encryption* introduces encryption[8] onto an individual field. It does not replace Encryption At Rest (EAR). It is a supplement to that feature. Developers invoke it in any of several ways. There are several functions introduced in FileMaker® Pro 16 with Field Level Encryption:

- *CryptEncrypt*
- *CryptDecrypt*
- *CryptEncryptBase64*
- *CryptDecryptBase64*

❖*Safeguarding Encryption Keys and Encryption Passwords.*

The introduction of these Encryption functions raises the question of how the keys and passwords should be generated, stored, protected, and (perhaps ultimately) deleted. This is particularly true for Field Level Encryption and for the Encryption Password used in Encryption At Rest. This topic has received almost no attention in the FileMaker Developer community. It is something we are going to need to review and to address to derive some recommended Best Practices. Losing the key is tantamount to forever being locked out of the file. This is a challenge for all Information Security systems, not just for the FileMaker Platform.

—Continues Next Page—

[8] https://fmforums.com/blogs/entry/1721-version-16-brings-major-new-security-features/

◆Putting Them All Together.

So, when we put all these elements together, we have the ***FileMaker Platform Security Building Blocks.*** To be sure, there are any number of other security-related items in any individual solution and in its deployment. But these are the foundation on which developers build.

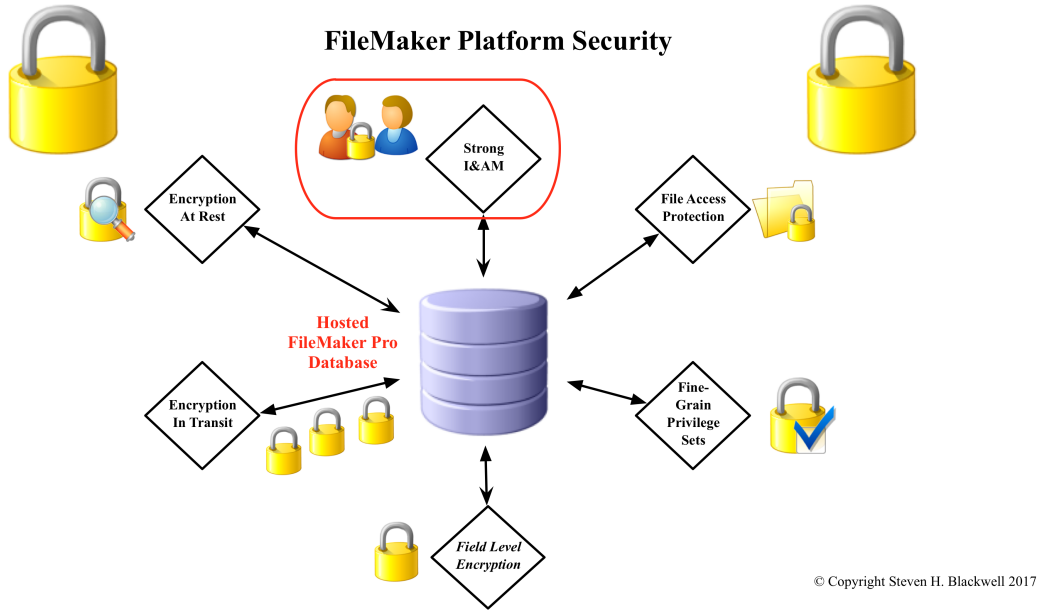

Figure 13. *FileMaker Platform Security Infographic*

—ACKNOWLEDGMENTS—

Knowledge gained over the course of a professional career springs from many founts. Clearly that is the case with FileMaker Platform Security. That said, I am deeply indebted to renown FileMaker Developer **Wim Decorte** for the information and guidance he has provided to me and to the FileMaker Developer Community for the better part of two decades. He truly is the "Developer's Developer" as many call him.

Likewise, I am indebted to **Stephen Dolenski** for his assistance at FM Forums [www.fmforums.com] in publishing and publicizing many of my White Papers and for hosting my FileMaker Security BLOG.

I am also indebted to **Barbara Levine**, well-known FileMaker Developer, for her assistance in reviewing this White Paper prior to publication. Both her keen eye and deep understanding of FileMaker Pro made this a better paper.

—ABOUT THE AUTHOR—

STEVEN H. BLACKWELL is a FileMaker Business Alliance Platinum Member Emeritus. He is the author of *FileMaker Security: The Book* as well as numerous White Papers and Technical Briefs about FileMaker Platform Security. He is also the creator of the FileMaker Security BLOG (http://fmforums.com/blogs/blog/13-filemaker-security-blog). He has over twenty-five years of experience with the FileMaker Platform going back to its earliest days. He is the President and CEO of Management Counseling Services.