Report : Disabling Security Settings in Acrobat Files...

Objectives:

- Demonstrating Saving PDF File as PostScript.
- Redistilling With No Security Settings Enabled

The encryption settings in Adobe Acrobat files can be disabled completely with relative ease, destroying the ability to protect intellectual property. This has extreme implications for authors and publishers who wish to protect their work. All that is needed is **Mac OS X** and Mac OS X's **Preview** application.

It has been suggested by Adobe that this is a Mac OS X issue, and has nothing to do with Acrobat at all. I must disagree completely. Suppose that a flaw is discovered in Internet Explorer's SSL security model. Now, let's say that a fictitious program called "extract" is used to exploit this security hole. Is this the fault of "extract", or is it the fault of Explorer? Obviously, the answer is Explorer. In this report, I show a similar scenario. Mac OS X's Preview program is able to ignore the security settings in an Acrobat encrypted file and do whatever it wants with the file. And if OS X's Preview can do this, then any program can be written to exploit this security hole. I would hope that Adobe takes the time to read this report, and acts quickly to correct the issue.

Disabling Acrobat Security...The Process

The process of destroying the security settings in an encrypted PDF document is surprisingly easy and straightforward. In my tests, I used Acrobat Distiller 5.05 running on Mac OS 9.2.2 to create a PDF file with very high security settings, locking out the ability to print, copy, or extract content of any kind.This is shown in Figure 1. The following steps detail how I was able to circumvent these security measures.

Figure 1-1:- The test document was distilled with high encryption settings. As shown here, no changes are permitted, and printing is disabled.

1. Open the encrypted file in Mac OS X **Preview**, located in the Mac OS X **Applications** folder.

2. Select **Print** from the **File** menu.

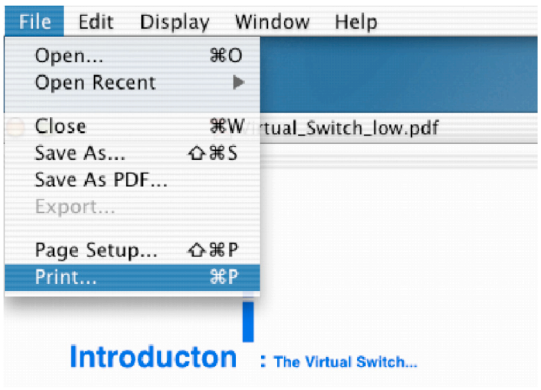

Figure 1-2:- When the encrypted PDF is loaded into Mac OS X Preview, the Print command is available, when it should not be. Recall that we completely locked out printing when the job was originally distilled.

3. When the standard print dialog pops up, select **Output Options** in the drop down menu.

4. Select **Save as File**, and choose **Postscript** in the **Format** menu.

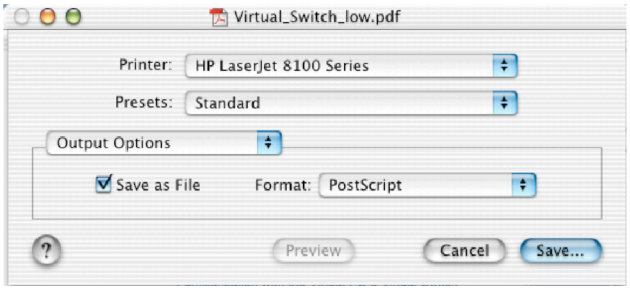

Figure 1-3:- Here's where all traces of PDF security are thrown into the proverbial recycle bin. Raw Postscript does not support the custom encryption technologies found in PDF and saving the file as such destroys all encryption present in the original PDF.

5. Once the file is saved is Postscript format, it can be re-distilled with new job options that permit printing, extraction of content, and general editing of the document. I set up such a folder in Distiller and re-distilled the new Postscript file. After performing a **Document Security** query in Acrobat on the new PDF file, the results show no trace of the original PDF security options that were supposed to protect the original file from alterations and printing.

Figure 1-3:- With all PDF security settings erased, I now have full access to the file to do with as I please, something the PDF security was supposed to prevent.

Summary...

Throughout this document, I have shown how simple the process is to remove all security from an Acrobat PDF file. It is my hope that Adobe fixes this problem, and does not take the tack of passing the responsibility onto Apple Computer. Clearly, if Mac OS X can circumvent PDF security, then any other third party PDF solution can as well. And if this is the case, Acrobat encryption and security features are useless.

If Adobe requires any further information, I can be reached at the following:

Marc Hoffman
Editor/Webmaster, The OS Emulation HomePage
http://www.kearney.net/~mhoffman
emulator@mac.com

Update...

It appears that this tutorial has gotten some attention at Adobe. On March 8, 2002, I spoke with Kathi Rauth, Senior Product Manager with Adobe Systems. After a 20 minute conversation, several key points were established:

1. Adobe is aware of this situation.
2. Adobe has based their security model as such that the secured PDF file trusts the program viewing it to follow certain guidelines. If these guidelines are not followed, then security can be compromised, as I have shown here.
3. Adobe is, and has been for quite some time, in contact with Apple Computer regarding this issue. They are both working to correct this issue.
4. I brought up the point that working with Apple Computer is all well and good, but what if some other programmer or company comes up with a similar program that does the same thing? I pointed out that this could be a full-time job trying to put out that many fires. I suggested that a better approach would be to make all PDF viewers dependent on the PDF file, and not the other way around.
5. Adobe has inserted some legal "entries" into the PDF specifications binding developers to keep on the proverbial road. This in theory ensures that all PDF viewers stick to the proper specifications for viewing PDF files. I pointed out that hackers are not going to pay attention to legal entries, and that this fact alone puts authors' intellectual property at risk of being altered or redistributed/printed without their permission. I suggested that the solution to this problem lies in the technical realm, and not the legal realm. Adobe countered that both legal and technical solutions are required to fix the problem, and are probably wise to think so.
6. Adobe assured me that they have instructed their employees to try and be more helpful in telling authors how to better protect their PDF work, and to not pass the buck. ***The best way to do this is to password protect the document, requiring the user to type a password to even view the PDF***. That is precisely what I have done with this document.
7. Adobe assured me that it is working on finding a legal and technical solution to this problem, which gave me cause for relief that someone at the company is listening. I was assured that Adobe is listening.