The danger of metadata [digital footprints]

[Underground Security Paper no. 3] THE DANGERS OF METADATA BY: DIzzIE [antikopyright 2008]

This is the third Underground Security Paper designed to further empower you to give yourself some semblance of electronic privacy. If you haven't done so, go read over USP no. 1: Encrypting your Instant Messaging Conversations (http://forum.rorta.net/showthread.php?t=576) and no. 2: Encrypting Email Communiques (http://forum.rorta.net/showthread.php?t=1273).

What is metadata?

To put it bluntly, metadata is hidden data that can fuck you over. Fuck you over real hard and rough like, savvy? Often defined as "data about data," metadata is information about a specific file that's often included within the file itself but that's often not readily visible or modifiable to the end-user when z is viewing the file in the standard application that z would typically use to view the file. In other words, metadata provides background information about a file. Chances are that every document you create, every digital photograph you take, every music file you download, and so on, all have little bits of metadata which can leak vital information about your identity.

What kind of data can metadata contain?

Embedded metadata can contain everything from your full name to device serial numbers to even your GPS coordinates. Other more mundane, but no less salient, bits of metadata may include the date the file was created, when it was last modified, the names of all the different personages who contributed to it, what applications or appliances were used to create the file, and so on. Suffice it to say that metadata is data that you will either want to delete entirely, or better yet, inject false data so as to spread disinformation (see How to Lie to People (http://forum.rorta.net/showthread.php?t=895) for a more in-depth look at this general strategy).

The rest of this textfile will discuss five examples of metadata in three different common file formats (DOCs, PDFs, and JPGs), as well as Thumbs.db files and MRU file lists (which we'll get to later on), along with describing how you can modify that data as well as taking preventative measures to make sure your data isn't accidentally leaked in the first place. Be sure to read the entire textfile even if you don't give a flying fuck about a particular file format as potentially valuable vignettes and notes are sprinkled throughout.

PDF Files (not to be confused with pedophiles)

The metadata in PDF files can store everything from your full name to the name of the application that the PDF file was created and/or edited with.

Let's say you just finished working on an important file in Word and proceed to print it to PDF using Adobe's PDF printer driver. You may not remember that when you first installed Word you just so happened to register it in your real name, or maybe your parents did, or maybe the

library that you're typing the document in has it registered under their name. Surprise, surprise, that name that Word is registered in (which is also set as the default Author) gets passed along into the Author field of the PDF's metadata.

Open up the PDF in Adobe Acrobat and hit Ctrl-D to access to the Document Properties window. You'll now see all sorts of fun data, some of which you can edit from within Acrobat Professional (namely Title, Author, Subject, and Keywords), but you will also see other data which you will be unable to edit, at least not from within Acrobat (the file creation/last modification date/time, the version and name of the program used to create the PDF).

Nota Bene: Acrobat Reader doesn't even let you edit the partial data that Acrobat Professional does, so you will need to use one of the third party programs discussed below. If you have Acrobat Pro, however, you will still only be able to modify or delete some of the metadata by either editing it in the Document Properties dialog or by going to

Advanced–>PDF Optimizer–>Discard User Data–>Discard document information and metadata–>OK.

There is a freeware program by the name of PDF Info (http://www.bureausoft.com/pdfinfo.exe) which lets you edit not only the aforementioned Title/Author/Subject/Keywords fields, but also the PDF Producer and Creator Application fields. It doesn't, however, let you change the file creation and modification dates and times.

In order to modify the dates and times you'll need to use a hex editor to manually change the data yourself. A simple free hex editor for Windows is called HexEdit (http://www.expertcomsoft.com/download.htm) and will allow you to perform the changes you need to the PDF file that PDF Info and Acrobat don't allow you to (you can also always open the PDF file in Notepad, but this can take a while and will cause slower computers hang).

Download the free version of HexEdit, make sure the PDF file you want to edit isn't currently open in any PDF viewer or whatnot, and then open it in HexEdit (better yet, make a copy of the file and use the copy to practice editing the metadata on, just to make sure you don't fuck anything up). Press ctrl-F to bring up the Find window, and change the search type from the default Hex to ASCII. Put in 'created' and start searching through the file. Once you find the created date on the right-hand side, go to Edit–>Allow Changes (so as to turn off Read Only mode), and then highlight the date on the right-hand side, and type in your new fake date in its place (or delete the date altogether). Click Find Next to continue searching the file for 'created' as the metadata appears in the PDF file more than once. Then repeat your search again for the terms 'creation,' 'modified,' and 'modify,' and similarly either replace or delete the dates, once again being sure to repeat each search so that any potential multiple instances of the field can be located and modified or blanked out.

Save and close the PDF file in HexEdit, and open it in Acrobat. Hit Ctrl-D and look over the new created/modified dates. If the dates are the same as those in your original PDF file, it means that you didn't find and replace (or delete) all of the metadata.

Nota Bene: Remember to make sure that your forged dates make sense. In other words, don't pick outlandish years like 3010 or well known dates like 09-11-2001. Furthermore, make sure that your dates match up and are sequential. In other words, all instances of the creation date should match, including the time, and all modification dates should be later than the file creation dates, and likewise match up.

Keep in mind that at this point you've only changed the creation/modification dates found in the PDF's metadata. The file's external dates will need to be further modified. . To modify the external creation date of the file, modify your system clock to reflect your desired creation time (which should match the creation date you specified in the PDF), and then copy the PDF files to another folder (be sure to copy them, not cut or move, as neither of those will change the creation date). To change the modification date, run the files through Touch (http://www.dizzy.ws/Touch.zip), a light Python script written by Bitplane that will spoof the modification date at various intervals. Your creation and modification dates should now have been successfully changed to reflect the date/time you indicated in your system clock.

If by this point you're wondering why the fuck you should piss away all this time putzing over a few dates, consider our aforementioned example of the library. Let's say that you are typing up an anonymous communique from the library, and unbeknownst to you, the library's name gets embedded into the PDF file since that's the name their copy of Word was registered with. Once your PDF is forensically analyzed by the piggies, they'll see that it was composed at Dumbfuck Library at 23:23 on February 3rd, 2003 (incidentally, you should never spoof a date that looks like that, can you tell why?). Surveillance footage will then be examined at that library around that date and time, and all of the sudden your anonymous communique now has a face attached to it. When that footage is further linked to you walking outside to the parking lot, that face now has an address procured from looking up the license plate registration information. So yes, dates fucking matter.

Or if you prefer a less dramatic example, let's say you're submitting a report for work or school, and you submit it a few hours past the deadline. If your teacher complains, tell them the email servers or the submission form must be laggy, and try showing them the document creation dates as evidence. Or what if your fuck buddy finds pictures of you with another fuck buddy? Just show zir the file creation dates which then go towards proving that the pics were taken when you weren't together.

DOC Files

Microsoft Word file metadata is probably the most famous type of metadata due to all the news stories about dumbass politicians and fat cat capitalists and the like (http://www.nytimes.com/2005/11/07/business/07link.html?ei=5090&en=98e8af679a0797f4&ex=1289019600&pagewanted=print) fucking up and leaving damning metadata in their DOC files (I especially love the bit where an anti-P2P tirade allegedly authored by California's attorney general was found to be authored by a member of the MPAA). The data may include everything from the names of all the different authors who worked on the file, to lines of text and comments that have been
deleted in previous revisions of the document in question.

To reduce the amount of metadata in your DOC files, be sure that the Fast Save (Tools–>Options–>Save–>uncheck Allow Fast Saves and Background Saves) and Track Changes (Tools–>make sure 'Track Changes' isn't selected) options in Word are turned off, and that Word automates the deletion of at least some personal information (Tools–>Options–>Security–>enable 'remove personal information from file properties on save.'). You can also download the Remove Hidden Data tool plug-in (http://tinyurl.com/2qaax), which will automate the deletion of some metadata, but not any of the date/time stamps, which you'll have to modify manually by changing your system clock to reflect your desired time/date, and then opening the document in question and then saving it again (to spoof the last modified/saved dates), or pasting the contents into a new file (to spoof the file creation date). Finally, open up the DOC file in a hex editor (just like you did with the PDF file), and comb through it too ascertain that there is no extraneous metadata left floating about. And of course, the obvious third choice is to simply stop using DOC files.

JPG Files

Aside from the fact that JPGs can contain information about the program that they were created with (for instance, if the file says 'ducky' in the first few lines when opened up in a hex editor, it was created with an Adobe application–that or someone made it look like it was created with an Adobe application), the gravest danger of JPGs lies in those that have Exchange image file format (Exif) metadata (as well as other metadata), namely photographs taken either with a digital camera or with a camera phone (though not all camera phones currently embed Exif data into their images, this trend may soon be changing, as was the case with digital cameras years earlier).

The newer your digital camera is, the less privacy you have. Newer cameras leak everything from serial numbers to even the GPS coordinates of the camera's location when the photo was taken. Though don't worry, older cameras still leak plenty of metadata as well, ranging from the camera's model to the date the photo was taken.

Photo Exif data became hot news a little while back, when it was discovered that the person who uploaded photos of the seventh Harry Potter book didn't bother to clean out the Exif data (http://entertainment.timesonline.co.uk/tol/arts_and_entertainment/books/article2104250.ece?print=yes), thus leading to the discovery of the camera's serial number. If z had ever bothered to register the camera, or had ever sent the camera in for repairs or upgrades, then zir name and address would be easily traceable. Good thing that zir camera didn't have the geolocation capability .

There are shitloads of non-free programs which can provide you with a fancy GUI to edit or view your Exif data (PowerExif and Exif Farm come to mind, with PowerExif being especially useful in that it offers you both batch processing and plenty of suggestions of different variables you could replace existent ones with, for instance different model names/numbers), but the job can be done using free software, with only a slightly higher learning curve. Now while I haven't been able to find a free program with a candy-assed GUI that can handle both batch editing and removal of metadata (though feel free to poke around yourself http://www.photo-freeware.net/exif-data-tools.php) there is a command line utility that does the job quite well.

Nota Bene: If you just want to remove all Exif data from a set of photos, you can even more easily run them through the GUI-based (and aptly titled) Exif Tag Remover (http://www.rlvision.com/exif/about.asp).

If, on the other hand, you want to tweak your Exif data to report spoofed information so as to fuck with anyone who may want to track you, you'll need to use the command-line ExifTool (http://www.sno.phy.queensu.ca/~phil/exiftool/) (there's also a basic GUI interface (http://freeweb.siol.net/hrastni3/foto/exif/exiftoolgui.htm) available for ExifTool, which you can try playing around with if you prefer that to the command line).

The first thing you'll want to do is get a read-out of all the Exif data the image contains. Download the zip file with the latest version of Exif tool, extract the file exiftool(-k).exe somewhere, and drag a sample JPG photograph onto it. A command-line window will pop up which will display all of the available data. If you want to output the data to a textfile, make another copy of exiftool(-k).exe and rename it to exiftool.exe. Next, click on Start–>Run–>type 'cmd' to bring up the command prompt. Type 'cd "directory where exiftool.exe is" (for example, cd "c:\program files\exiftool"), and then type: exiftool "file path of your image or folder of images" > info.txt (making sure that there is no trailing slash at the end of the directory or file path, i.e. "\my photos" instead of "\myphotos\") and you should get a read-out of the available metadata in a file called info.txt in the same directory that exiftool.exe is located.

After you see all of the available data you can start picking which data you'll want to modify (preferably the camera make, model, serial number, GPS coordinates, software, and all of the date/time fields). Alternatively, if you just want to delete all the metadata and don't want to use the aforementioned Exif Tag Remover, you would simply type: exiftool -overwrite_original -all= "file path to either the folder or the image to clean". Once you find the fields that you would like to modify, you'll need to look-up the tag name (http://search.cpan.org/~exiftool/Image-ExifTool-7.21/lib/Image/ExifTool/TagNames.pod) and then proceed to craft a command that will modify all of the pertinent fields.

Here's a sample command you could execute:

exiftool -overwrite_original -make=moo -model=poo -software=goo -cameraserialnumber=2323 -alldates-="0:2:3 5:0:0″ "C:\whatever\myphotos"

This command will overwrite the original photos, change the camera make (the brand), the model, and the camera software name (which can reveal the camera brand), as well as modify the serial number and move all the dates in the Exif data back two months, three days, and five hours. Some cameras use the 'serialnumber' tag instead of 'cameraserialnumber', so if you receive an error in ExifTool, try the other tag.

Nota Bene: While the ExifTool command discussed above will modify all of the dates found within the Exif data fields of the image, it will not modify the actual file creation/modification date. To modify the creation date of the file, modify your system clock prior to copying over the photos from your camera or phone. If you already copied the files over, go ahead and copy them to another folder (be sure to copy them, not cut or move, as neither of those will change the

creation date), and then run them through ExifTool. Your creation and modification dates should now have been successfully changed to reflect the date/time you indicated in your system clock. To change the modification date without running ExifTool, run the files through Touch (http://www.dizzy.ws/Touch.zip), a light Python script written by Bitplane that will spoof the modification date at various intervals. (If you've been reading the entire text, this procedure should be ringing a bell, as it's the same thing you should have done to modify a PDF file's time/date stamps as well, the same procedure works for any other file).

By now you should have a nicely spoofed series of photos, but why stop there? There are a variety of programs available that will allow you to insert GPS coordinates into the photo's metadata (a recent fad that's been dubbed 'geotagging' that we can use to spread a wee bit of the old disinformation). Grab the free PhotoMapper (http://software.copiks.com/photomapper/), and input the custom latitude/longitude coordinates you want, and then press 'Tag selected images.' If you now open your spoofed photos in our old friend ExifTool, you should see brand spanking new GPS metadata fields complete with your bogus coordinates .

If you need to get the GPS latitude/longitude coordinates to inject into the image, head on over to Google Maps (http://maps.google.com), find a location you want the photos to appear to be from, and click the 'Link to this page' link in the top-right corner.

Copy the URL that appears and you should see a &ll=23.2323,46.4646 variable in the URL.

The first number is the latitude and the second is the longitude. Plug those into PhotoMapper and hit 'Tag selected images.' Your photos should now have the spoofed GPS coordinates in them .

Thumbs.db

Whenever you view files as thumbnails in Windows (View–>Thumbnails), a hidden Thumbs.db file is created which stores the names of the files and a small thumbnail image of all of the files in the folder, so long as they are photo or video files. Even after you delete or move the files from that particular folder, the Thumbs.db file retains the thumbnail version of all images that were in that folder. Ever send a folder of images to someone, deleting any files you don't want them to see? Well, they can still see them by using a free Thumbnail Viewer (http://www.itsamples.com/software/tdv.html).

Since the Thumbs.db file is a hidden system file, you need to enable viewing hidden files in order to be able to locate it. Open up any file folder, and go to Tools–>Folder Options–>View–>select 'Show hidden files and folders' and uncheck 'Hide protected operating system files.' Now simply drag the Thumbs.db file onto Thumbnail Viewer, and you'll see all of the thumbnails and filenames imbedded in the db file. In order to disable this grave privacy violation, go to Tools–>Folder Options–>View–>check 'Do not cache thumbnails' (a feature that's insanely enabled by default).

Nota Bene: In Windows Vista, the thumbs file is no longer stored in each folder but is instead saved in a centralized location:

%sysroot%\Users\%profile%\AppData\Local\Microsoft\Windows\Explorer\, with each file being called thumbcache_xxx.db, where xxx is a varying number.

Finally, to delete all of the Thumbs.db files, either use the Thumbnail Database Cleaner (http://www.itsamples.com/software/tdc.html), or simply go to Start–>Search–>For Files or Folders…–>All files and folders–>put in 'Thumbs.db' in the 'All or part of the file name' field–>select the location to Look in: (it's best to scan all of your drives)–>hit Search. Then just delete all of the found results.

MRU Files

Finally, while Most Recently Used (MRU) files aren't often mentioned in metadata discussions, they most certainly fit the metadata definition of being "data about data," so a brief note on them is in order. MRU files contain lists of the most recently viewed files in a wide array of applications (from word processors to media players). Lucky for us, there is a free, easy to use program (which should be a welcome sight after dealing with ExifTool), MRU-Blaster (http://www.javacoolsoftware.com/mrublaster.html), which will scan your drives for a wide variety of MRU file lists and then delete them all.

Wrapping Up

If it isn't fucking obvious by now, metadata is highly dangerous. If you're not careful it can lead not only to potentially embarrassing situations but can also be used as forensic evidence against you for whatever reason. The least possible course of action you should undertake, particularly if you're pressed for time, is the outright deletion of all available metadata in your files. If, on the other hand, you have some time to kill, it would behoove you to go ahead and forge all of the data to your advantage. Make it look like you used a different camera, operating system, and software application on a different date at different time.

Also keep in mind that while most of the examples in this text have been fairly Windows-centric, metadata is a significant problem on all operating systems, with similar tools likewise existing for different OSes that do similar jobs to the ones discussed in this guide. In other words, don't think that just because the guide mainly discussed Windows tools that the problem of metadata doesn't apply to you.

Stay sharp, and keep your head down. As Freddy N once wrote,

If you don't want your eyes and mind to fade, Pursue the sun while walking in the shade.

And once again, be sure to check out the two earlier textfiles in the Underground Security Paper series:

USP no.1: Encrypting Instant Messaging Conversations (http://forum.rorta.net/showthread.php?t=576)
USP no. 2: Encrypting Email Communiques (http://forum.rorta.net/showthread.php?t=1273)

For more knowledge check out www.rorta.net & www.dizzy.ws. Send comments to xcon0 @t y@hoo d/0|t c\0|m.