Leaking Sensitive Information in Complex Document Files—and How to Prevent It

Simson L. Garfinkel | Naval Postgraduate School

Complex document formats such as PDF and Microsoft's Compound File Binary Format can contain information that is hidden but recoverable, as a result of text highlighting, cropping, or the embedding of high-resolution JPEG images. Private information can be released inadvertently if these files are distributed in electronic form. Simple experiments involving the creation of test documents can determine whether a particular program embeds hidden information.

In April 2011, the UK Parliament posted an Adobe Acrobat PDF file on its public website detailing key vulnerabilities of UK nuclear submarines. Portions of the document had been removed by an official at the British Ministry of Defence, but the redactions were made incorrectly: the sensitive text was simply obscured with black boxes and could be recovered by copying the text from the PDF and pasting it into a word processor. A follow-up investigation by *The Daily Telegraph* found similar PDFs with secrets covered by black boxes on four other UK government websites.[1]

Leaks of sensitive information in PDFs are frequently newsworthy, but hardly new. In October 2003, the US Department of Justice provided a report about its internal efforts to increase racial diversity in response to a Freedom of Information Act request, with each of the report's embarrassing findings and recommendations blacked out. Journalist Russ Kick discovered that the black boxes could be removed easily[2] and posted an unredacted version of the report on his website.

"I was kind of surprised," Kick told *The New York Times*, "but we are talking about a government bureaucracy, so I wasn't that surprised."[3]

But it's not just government bureaucracies that are at fault. The *Times* had made the same mistake when it published an alleged US intelligence report from the 1950s on the paper's website in 2000.[4] *The Washington Post* made a similar mistake in 2002 when it posted a scan of the DC sniper's demand letter.[5]

Such failures might prove more common in the coming years as healthcare organizations, law firms, other businesses, and even users at home distribute information in electronic form after first attempting to remove sensitive financial or health information. These failures result from the complexity of these file formats, including PDF and Microsoft Compound File Binary file format (MS-CFB) used by the Microsoft Office suite, interacting with the complex software used to create and redact these files.

Fortunately, there are solutions available for redaction, and there are good design patterns that software developers can implement to make such privacy failures less likely in the future.

PDF Privacy Leaks

In the 1990s, Adobe created PDF, an electronic file

Copublished by the IEEE Computer and Reliability Societies 1540-7993/14/$31.00 © 2014 IEEE

format that would let any modern computer view or print any document without needing to install software or fonts (provided that a general-purpose PDF reader was installed). Adobe published the PDF specification and licensed the patents required for implementation. PDF soon became a common format for distributing born-digital documents, scans of paper documents, and even electronic forms. Its popularity came in part because PDF could distribute documents without the malware risk associated with the Microsoft Office file format.[6] Today, numerous programs can create, display, and even edit PDF files, and the format has been adopted as an international standard (ISO 32000-1:2008).

PDF is based on Adobe's PostScript Language, but it's more specialized. Like PostScript, PDF has commands for drawing lines, embedding fonts, and displaying text. PDF also has direct support for document structure, a variety of image and video compression algorithms, electronic forms, digital rights management, and cryptography. PDF even allows embedded JavaScript for form validation and other kinds of automation.

PDF's rich support requires a complex reader, and such readers inevitably have flaws. Many of these flaws have been exploited by attackers using cleverly constructed documents.[7] But, whereas PDF exploits depend on implementation errors, PDF privacy leaks result from properly formed files containing information that's invisible but readily extracted.

Creating PDFs

Users typically create PDF files in one of two ways. One way is to generate a PDF directly from an application program by "printing" or exporting. PDF files created in this manner contain the actual characters and fonts rendered by the output device and have sharp, crisp letters; have text that can be selected and copied; and are searchable. These PDF files are considered *accessible*, because people who are vision impaired can readily access the documents by enlarging the text or using screen readers.

A second way to create PDF files is to scan a paper document page by page, producing a PDF file that's essentially a collection of photographs. Even when scanned at 300 dpi, the resulting document can appear pixelated, mushy, and difficult to read on a computer screen, because the PDF reader is displaying a photograph of text rather than rendering the characters. (Rendering allows the use of digital typographic techniques such as font hinting and antialiasing that significantly increase legibility.) Such text isn't accessible—it can't be selected, copied, searched, or read with a screen reader. These files tend to be 10 to 100 times larger than PDF files created with the first method. Nevertheless, many PDF files in circulation are scans of paper documents because the originals weren't prepared using a computer, the original electronic files are no longer available, or the individuals who prepared the files didn't have the necessary tools or training to create an accessible PDF file.

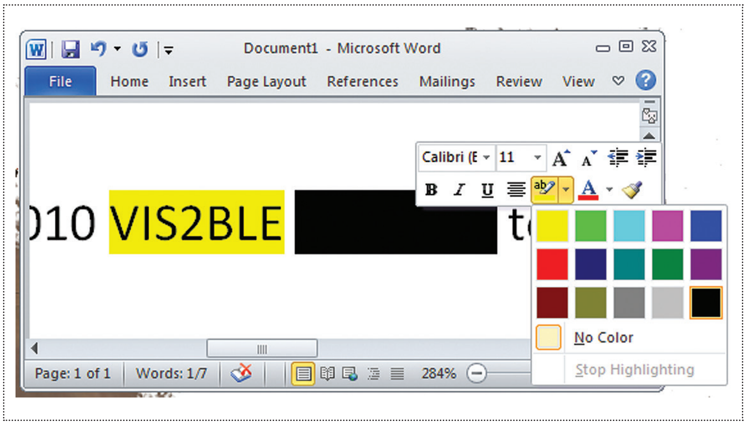


Figure 1. Microsoft Word 2010's Text Highlight tool can highlight text with different background colors. When the background is black, the text appears to be redacted but is still present.

Adobe's Acrobat Pro and other PDF processing programs can "recognize" text in scanned PDFs with an optical character recognition engine. Acrobat Pro can then modify the PDF file and place the recognized text in a second layer underneath the scanned image. The result is a scanned PDF that's searchable but that preserves the original image. The recognized text can even be copied into another program—an example of hidden text with a genuinely useful purpose!

The PDF standard doesn't distinguish between accessible and nonaccessible PDF files. From the point of view of the standard, a PDF is simply a file that contains objects (for example, text drawing commands and fonts), a file structure that describes how and where the objects are stored in the file, a document structure that describes how the objects are used to create pages, and one or more content streams that describe how to display pages on the screen and how to print them. PDF's power comes from its ability to compose these features in many different ways.

Textual Privacy Leaks

Many of the PDF-enabled privacy leaks reported in the media appear to result from the improper use of Microsoft Word's Text Highlight tool to hide information (see Figure 1). The interaction of features that results in privacy leaks isn't specific to Word: it's exhibited in all but one of the word processors tested for this article (Word, Apple Pages, Google Docs, and LaTeX). The standout was LibreOffice, which avoids the problem somewhat,

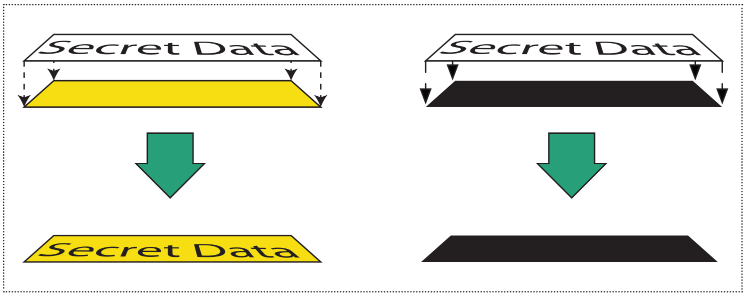


Figure 2. The Text Highlight tool places a colored box underneath the text. When the box is the same color as the text, the text appears to be absent but is still present.

although it can still be manipulated to produce a PDF with hidden information.

In normal use, highlighting turns the background color of text from white to yellow to attract attention, similar to highlighting with a yellow marker. But word processors don't use translucent ink. Instead, they highlight text by first printing an opaque colored box, then printing black text on top of the box.

Privacy leaks can happen when the user changes the highlight color from yellow to black, resulting in black text printed on a black box. When the text is printed, Word still generates instructions to print both the box and the text (see Figure 2). The PDF file captures these instructions, and both can be recovered independently.

This behavior is particularly pernicious because the resulting text looks very similar to a document that has been redacted through the official procedures used by many governments—obscuring sensitive text with black boxes. Furthermore, there would likely be no significant privacy leak if the document was printed and distributed on paper. It's only the distribution of the PDF created directly from the word processor with black boxes under black text that results in the privacy leak.

The easiest way to recover the hidden text is with Acrobat Reader's Copy and Paste commands: copy the blacked-out text and paste into a word processor. A more dramatic demonstration is to open the PDF with an editing tool such as Adobe Illustrator or Inkscape. These tools understand PDF files' internal structure and allow objects on each page to be manipulated individually. Once in the editor, the user can select the background box and change its color from black to fuchsia (see Figure 3), resulting in the highlighting of text that was intended to be redacted.

"Highlighting" text with black isn't the only source of textual privacy leaks. In the case of the redaction errors at *The New York Times* and *The Washington Post*, both papers apparently scanned a paper document, then drew black boxes on the scanned pages using Adobe Photoshop. By default, Photoshop draws boxes in a new layer, allowing the boxes to be moved after they're drawn. The PDF format preserves these layers, allowing recovery of the original bitmaps. Of course, recovery isn't what the editors intended. Both newspaper leaks would have been avoided if the Photoshop operator had "flattened" the layers before the PDF was made available on the newspaper's Web server.[5] Publishing material as a PNG or JPEG would also have avoided the leak, as those formats don't support multiple layers.

Image Privacy Leaks

Private information can also leak in PDFs with a high-resolution image or sensitive metadata. Thus, organizations and individuals that make PDFs containing JPEGs available for download should understand how PDFs are created and analyze them prior to release to ensure that the JPEGs don't contain undesired information.

For example, when a Macintosh user drags a JPEG digital photograph from the desktop into a Word Mac 2011 window and saves the file, Word embeds an exact copy of the JPEG inside the resulting Office Open XML (.docx) file. The .docx file is a ZIP file, and the original JPEG can be readily extracted using any "unzip" utility. Likewise, when Word produces a PDF from the document, it embeds the original JPEG directly in the PDF as an object, along with instructions to scale, rotate, and possibly crop the JPEG. The original JPEG can be extracted using Acrobat Pro or an open source tool such as pdfimages.

High-resolution JPEGs in PDFs can be both annoying and risky. The annoyance is that the files are typically much larger than needed: a typical high-resolution JPEG can be 3,000 pixels across and 3 Mbytes or larger in size. If the image is shrunk to two inches, the result is a super-resolution image with 1,500 dpi of information (only 150 dpi are required for photographic-quality production).

The privacy risk is that super-resolution images can reveal information without the knowledge of the document's author, editor, or publisher. People in a photo who are too small to distinguish might be clearly identifiable if the photo is enlarged. High-resolution photographs of a person's face can be used as a biometric. License plates, computer screens, and sometimes even papers on a desk can be enlarged and read.

Embedded JPEGs can also contain Exchangeable image file format (Exif) metadata. Exif information can include the model and serial number of the camera that took the picture, the date and time it was taken, and GPS coordinates.

My analysis of desktop applications for this article found that they were inconsistent in the way they embed JPEGs. Both Word and Apple Pages for Mac embed the unmodified JPEG in document files and

PDFs generated from the Print menu. But when users invoke Pages' Export PDF command, they can specify a quality of Good, Better, or Best. Choosing Good or Better causes an embedded JPEG to be converted to a lower-resolution image, removing the metadata in the process. Likewise, both Microsoft PowerPoint and Apple Keynote reduce the resolution of high-resolution JPEGs embedded in slide presentations exported as PDFs. Overall, it appears that the metadata stripping is an inadvertent side effect of creating a lower-resolution JPEG, rather than an intentional effort by application authors to strip privacy-sensitive information.

Cropping Privacy Leaks

Cropping, masking, and rubberbanding can also result in privacy leaks, because many tools implement these operations by embedding the original object in a document file, then applying a postprocessing operation. As is the case with hidden-but-present text, special tools can undo the transformations and reveal the original content.

Programs are also inconsistent in alerting the user to the risk. Apple Preview 6.0.1 (distributed with MacOS 10.8) displays an alert box warning "the content outside the selection is hidden in Preview, but you might be able to view it in other applications" when cropping, but not when a rectangular selection is rubberbanded on a PDF page and the object is copied and pasted into another document. However, both the cropping and the pasting operations result in the entire original PDF page being embedded in derived documents. Although evaluating the PDF commands and removing those that execute outside the displayed area is possible, such an implementation is significantly more complex. Adobe Acrobat XI Pro provides no warning at all.

To avoid this privacy leak, crop the image in a bitmap manipulation program, such as the GNU Image Manipulation Program or Adobe Photoshop.

A PDF Privacy Experiment

To test whether word processing software will embed black-on-black text into a PDF, create such a file in a word processor, produce a PDF, and attempt to find the hidden text in the resulting file. Privacy-conscious users should test their tools to determine when the tools leak private information because software changes on a regular basis—far too fast for academic literature to keep up.

For this article, I created six test documents using Microsoft Word (Mac and Windows), Google Docs, Apple Pages, LibreOffice, and LaTeX. Each document had a single line of text consisting of a number, the name of the product, the word "VIS#BLE" with a yellow highlighted number, the word "HID[redacted]DEN" with a black highlighted number, and the word "test." (In this paragraph, I used the # symbol instead of a number, but in the experiment the number was changed for each word processor.) Thus, each file had two kinds of highlighted text—one that a naive user would expect to find in a file and one that such a user would not—and each instance of highlighted text was different, allowing the provenance of each piece of hidden text to be tracked if all were combined into a single file.

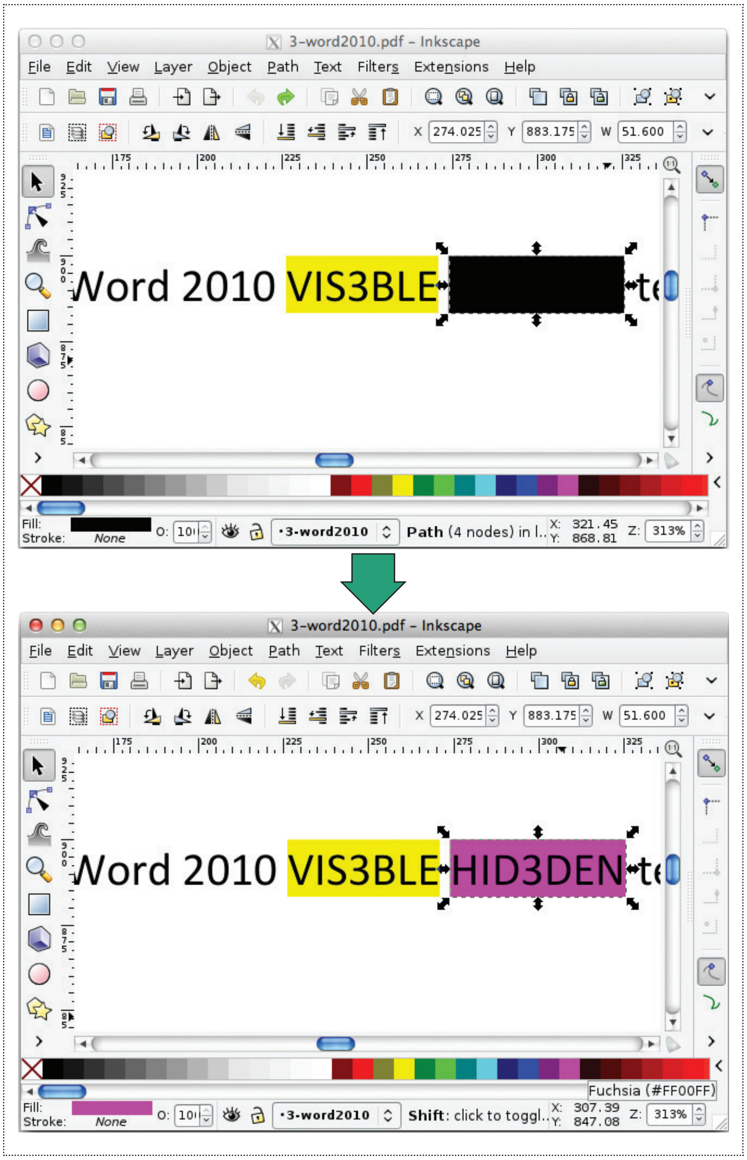
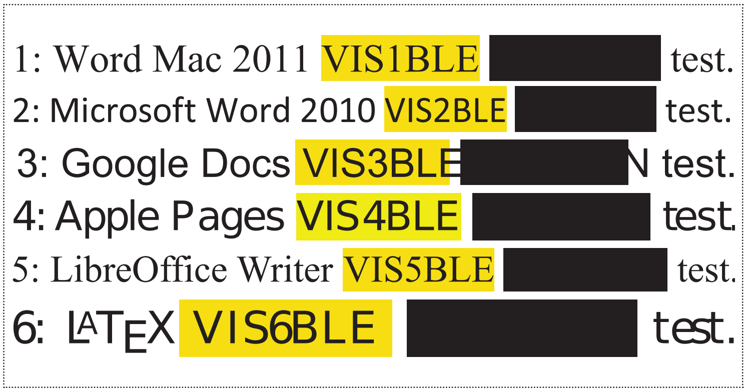

3-word2010.pdf – Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X 274.025 Y 883.175 W 51.600
Word 2010 VIS3BLE
Fill: Stroke: None
O: 100
•3-word2010
Path (4 nodes) in l..
X: 321.45 Y: 868.81 Z: 313%
3-word2010.pdf – Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
X 274.025 Y 883.175 W 51.600
Word 2010 VIS3BLE HID3DEN
Fuchsia (#FF00FF)
Fill: Stroke: None
O: 100
•3-word2010
Shift: click to toggl..
X: 307.39 Y: 847.08 Z: 313%

Figure 3. Black boxes created with the Text Highlight tool can be changed to a different color with PDF editing tools such as Inkscape.

All the programs I tested embedded highlighted text in the PDF file when the text was the same color as the background (see Figure 4). Readers are invited to analyze a PDF of this article downloaded from the IEEE website to see if the text is still present. This poses an important question: What is the correct behavior—to



1: Word Mac 2011 VIS1BLE [redacted] test.

2: Microsoft Word 2010 VIS2BLE [redacted] test.

3: Google Docs VIS3BLE [redacted] test.

4: Apple Pages VIS4BLE [redacted] test.

5: LibreOffice Writer VIS5BLE [redacted] test.

6: LATEX VIS6BLE [redacted] test.

Figure 4. Using the Text Highlight tool to create redacted text. (The misalignment of the Google Docs highlight appears to be a bug in Google's software.)

faithfully place the text and the same-colored box in the PDF file (what the user instructed), or to suppress the text (what the user sees on the screen and the printed page)? We will return to this question later.

Document Metadata Privacy Leaks

Metadata has come to mean data that describes other data, such as a title describing a document, a phone number describing a recorded audio call, or a date describing a table of scientific measurements. Document metadata can be stored within or separate from the document. For example, file systems store extrinsic metadata, such as the creation and modification dates of each file.[8] Some document file formats also contain internal or intrinsic metadata.

Both PDF and Microsoft Office support intrinsic document metadata including title, author, producer, creator, creation date, modification date, and keywords as well as user-defined metadata. We can find additional metadata, including copyright strings, color space names, file names, and the OS on which the file was created, by examining the file with a hex editor.

Many metadata privacy leaks result from documents being made available in the Microsoft Office binary file format. The most significant leaks have resulted from embedded change tracking information.

Comments and Change Tracking Leaks

Microsoft introduced the ability to track changes in a document with Word 6 on Macintosh and Word 95 on Windows. Change tracking must be specially enabled on a per-document basis and is used widely. Once enabled, Word records every addition, deletion, and formatting change; the time of each change; and the responsible user. Word also lets users insert nonprinting "comments" into the document.

With Word 2002, Microsoft introduced the ability to have "hidden" tracked changes, wherein the change tracking and comments are hidden from the screen and printed copy but the information is retained in the file. The changes can be viewed again by changing the file's display mode from Final to Final: Show Markup (or Final Showing Markup on Word for Mac).

As before, a complex document format that can contain invisible information combined with the lack of obvious indicators alerting to the presence of hidden data can result in the inadvertent release of sensitive information.

Perhaps the most monetarily significant case involving such information pertained to a Microsoft Word file for an article about the drug rofecoxib (Vioxx) that was submitted to *The New England Journal of Medicine* in May 2000. The article reported that prescription rofecoxib caused fewer stomach ulcers than over-the-counter naproxen in a study of 8,076 patients.[9] (The US Food and Drug Administration had approved rofecoxib the previous year.)

Four years later, rofecoxib's manufacturer Merck withdrew the drug from the market. A new study had shown that people taking rofecoxib had twice the incidence of heart attack or stroke as people taking a sugar pill.[10] Millions had been put at risk by using the drug, and an unknown number had died.

After Merck pulled rofecoxib from the market, the executive editor of *The New England Journal of Medicine* reviewed the magazine's paper files and discovered a floppy disk containing the original submitted version of the manuscript. The file's hidden change tracking revealed that a table detailing the same kind of cardiovascular events had been deleted from the article just two days before it was submitted to the journal[11] by a user named "Merck,"[12] implying that the company itself had suppressed the information.

If the table had been published in 2000, prescribing doctors would have known that Vioxx carried a risk of heart attack, and the drug might never have become so popular.

Merck ultimately paid US$4.85 billion in 2007 to settle 27,000 lawsuits by those claiming to have been injured by rofecoxib; $950 million more was paid to settle criminal charges resulting from the company's actions.[13] A key piece of evidence was that original MS-CFB file.

Other Office Metadata

MS-CFB files contain other information, including document author, keywords, the last time the file was printed, and path information for the previous 10 saves. Such metadata can reveal information that the document's author wishes to remain confidential.

For example, on 30 January 2003, UK Prime Minister Tony Blair's office released a Microsoft Word document entitled "Iraq—Its Infrastructure of Concealment, Deception and Intimidation," claiming it was a previously classified document detailing the case for invading Iraq based on the country's likely possession of weapons of mass destruction.

The Iraq document is somewhat infamous, because just days after its release, Glen Rangwala at Trinity College determined that the document was largely plagiarized from documents that were freely available on the Internet.[14] Based solely on the document's content, this analysis caused considerable embarrassment to the UK government.

But the Iraq document also leaked sensitive metadata. In June 2003, computer security expert Richard Smith showed that the actual MS-CFB file revealed the servers inside the Home Office where the file had been saved, and even when the document was copied to a floppy disk—reportedly so that US Secretary of State Colin Powell could have a copy for his 5 February 2003 presentation at the UN (www.computerbytesman.com/privacy/blair.htm).

A recent study of 15 million Microsoft Office files available freely over the Internet found that 97 percent included significant metadata, and 19 percent included all 10 revision histories.[15] The authors found that they could readily infer collaborators between corporations and the US military and could cross-correlate between document authors with Twitter accounts. "Our study raises major concerns about the risks involved in privacy leakage, due to metadata embedded in documents that are stored on public web servers," the authors concluded.

Solutions

All the privacy leaks described here resulted from the ability of complex file formats to contain hidden information that isn't obviously visible when files are created or viewed.

Broadly speaking, there are two approaches to address user interactions with computers that have potential security problems: train the users so that they don't perform those actions, or design systems so that performing such actions is unlikely or impossible.

Training

The computer industry has largely attempted to solve the problem of inadvertent disclosures by building manual redaction tools into some programs and training people to use those tools.

For example, Adobe Acrobat versions 8 and higher include a "redaction" tool. Acrobat redaction is a three-step process that involves first marking the portions for redaction, applying the redaction, and then optionally removing metadata from the PDF file. The resulting file has black boxes in place of the removed text, but unlike boxes applied in a word processor or Photoshop, there's no text underneath. Acrobat XI further allows users to annotate the black boxes with "exemption codes," as might be necessary when redacting documents released as a result of a US Freedom of Information Act request or when preparing documents that are partially protected by the US Privacy Act.

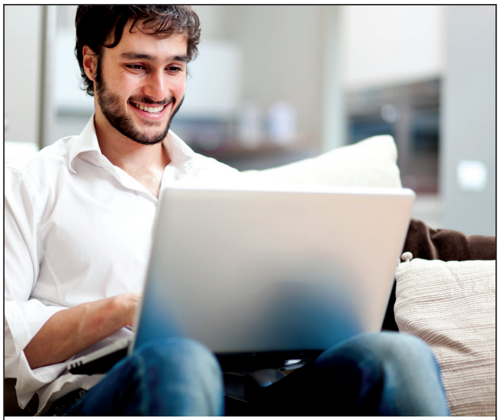
Likewise, Microsoft added a feature to its software to let users remove metadata. However, such removal must be performed explicitly. Microsoft's "Crabby Office Lady" advice columnist Annik Stahl instructed users to remove tracked changes, comments, and hidden text before sending out documents such as résumés, annual reviews, and contract bids (http://office.microsoft.com/en-us/help/track-changes-in-word-don-t-let-them-track-you-HA001139412.aspx).

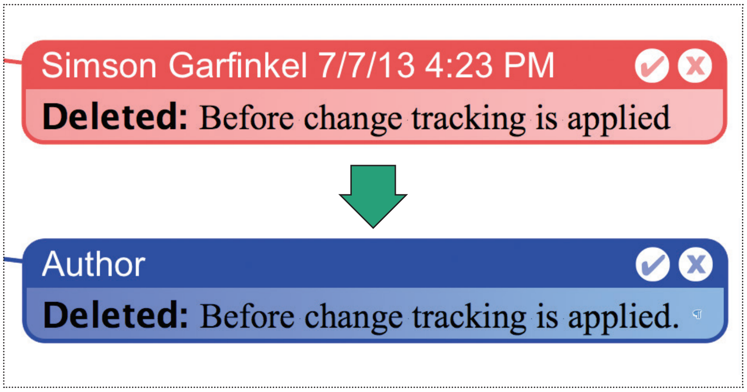
Finally, the US National Security Agency Information Assurance Directorate has published information on "Redacting with Confidence" that describes a variety of techniques for manual redaction (www.nsa.gov/ia/_files/support/i733-028r-2008.pdf).

Software Modification

An alternative to training is to modify word processors to make these kinds of redaction errors less likely or even impossible.

For example, when a user of LibreOffice Writer changes the background of black text from white to black, the program simultaneously changes the text color from black to white. This is implemented with a special text color called "automatic" that changes the text





Simson Garfinkel 7/7/13 4:23 PM

Deleted: Before change tracking is applied

Author

Deleted: Before change tracking is applied.

Figure 5. Microsoft Word's "remove personal information from this file on save" option removes the name of the person who changed the document but leaves change tracking information.

color "so the text is always distinguishable from the background" (www.libreoffice.org/get-help/accessibility). The feature is designed to promote usability but has the side effect of requiring two steps to create black-on-black text—one to set the background and one to change the text color. The developers could go further and simply disallow text and background to be set to the same color as a safety measure.

Another solution, which could be implemented by every word processor, would be for the programs to detect when a letter is being printed on a same-color background and suppress the letter.

LibreOffice helps address the problem of super-resolution images in PDF files by bringing the issue to the users' attention. LibreOffice's PDF export dialog lets users set the JPEG compression quality and specify a final image resolution in dpi. This is superior to Apple's PDF export dialog, which only allows users to specify an image quality of Good, Better, or Best but doesn't explain that these settings might impact document size or privacy.

Experiments with LibreOffice's PDF export function revealed that the program strips Exif information if it modifies the embedded JPEG but not otherwise. As before, the program's privacy functionality appears to be a side effect of features designed to prevent files from becoming unnecessarily large rather than an explicit attempt to limit the spread of potentially sensitive information.

The current versions of both Windows and Macintosh Word have privacy options for addressing sensitive metadata. One option, "remove personal information from this file on save," removes the name of the computer's user from document metadata. Tracked changes aren't removed, but the name is changed to "Author" and the modification time is removed (see Figure 5). A special Document Inspector claims to check for and remove 10 different kinds of hidden metadata, but the feature is buried beneath several layers of menus.

Disturbingly, the Microsoft Word user interface gives the impression that the program will also detect and eliminate hidden or invisible text. Tests show that these features don't discover text on same-color background. Instead, the feature identifies text with the Microsoft font property Hidden, which prevents text from printing. The apparent confusion results from Microsoft's use of the word "hidden" as a proper noun to describe a Microsoft feature, rather than as an adjective with its common meaning.

As a result of the metadata privacy incidents discussed in this article and others, several software vendors now offer enterprise metadata removal tools. For example, the PayneGroup's Metadata Assistant can be configured to remove metadata from Word and PDF files on demand or automatically, such as when files are sent by email to another organization. Metadata removal can be performed on users' computers or on a server. A typical customer might be a law firm that would configure its outbound mail server to automatically strip metadata from email attachments. According to the American Bar Association, 17 states now hold that attorneys have an ethical requirement to exercise "reasonable care" in removing metadata prior to transmitting documents.[16] Of those states, six hold that recipients of documents with metadata may be exploited, nine hold that such practice is ethically prohibited, and two hold that the situation is case specific. A 2010 survey by the American Bar Association found that 59 percent of the respondents' firms had some kind of specialized metadata removal software available, up from 46 percent the year before.

Both PDF and Microsoft Office files can contain information that's essentially invisible to the individual preparing the electronic document yet easily extracted by a recipient. Today, a significant number of documents available on the Internet for download contain significant metadata of all kinds. To address this problem, vendors and users have largely relied on increased training and specialty tools.

A better approach would be to modify tools so that the underlying data model is in line with what's presented in the user interface—that is, by making it harder for users to produce documents with hidden information. In the absence of such redesign, embarrassing data leaks are sure to continue. ■

References

1. D. Millward and T. Harding, "Secrets Put on Internet in Whitehall Blunders," *The Daily Telegraph*, 17 Apr. 2011;

www.telegraph.co.uk/news/uknews/defence/8457506/Secrets-put-on-internet-in-Whitehall-blunders.html.
2. D. Johnston and E. Lichtblau, "A Critical Study, Minus Criticism," *The New York Times*, 31 Oct. 2003; www.nytimes.com/2003/10/31/us/a-critical-study-minus-criticism.html.
3. T. McNichol, "Peeking behind the Curtain of Secrecy," *The New York Times*, 13 Nov. 2003; www.nytimes.com/2003/11/13/technology/peeking-behind-the-curtain-of-secrecy.html.
4. D. Johnston and E. Lichtblau, "Unediting the Editing of a Report," *The New York Times*, 31 Oct. 2003; www.nytimes.com/2003/10/31/us/unediting-the-editing-of-a-report.html.
5. R. Jones, *Internet Forensics*, O'Reilly Media, 2010.
6. R.M. Stallman, "We Can Put an End to Word Attachments," GNU, 2002; www.gnu.org/philosophy/no-word-attachments.html.
7. D. Stevens, "Malicious PDF Documents Explained," *IEEE Security & Privacy*, vol. 9, no. 1, 2011, pp. 80–82.
8. S. Garfinkel et al., "A Solution to the Multi-user Carved Data Ascription Problem," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 4, 2010, pp. 868–882.
9. C. Bombardier et al., "Comparison of Upper Gastrointestinal Toxicity of Rofecoxib and Naproxen in Patients with Rheumatoid Arthritis," *The New England J. Medicine*, 23 Nov. 2000; www.nejm.org/doi/full/10.1056/NEJM200011233432103.
10. R. Knox, "Merck Pulls Arthritis Drug Vioxx from Market," All Things Considered, 30 Sept. 2004; www.npr.org/templates/story/story.php? storyId=4054991.
11. G.D. Curfman, S. Morrissey, and J.M. Drazen, "Expression of Concern: Bombardier et al., 'Comparison of Upper Gastrointestinal Toxicity of Rofecoxib and Naproxen in Patients with Rheumatoid Arthritis,' N Engl J Med 2000;343:1520-8," *The New England J. Medicine*, 29 Dec. 2005; www.nejm.org/doi/full/10.1056/NEJMe058314.
12. R. Langreth and M. Herper, "Merck's Deleted Data," *Forbes*, 8 Dec. 2005; www.forbes.com/2005/12/08/merck-vioxx-lawsuits_cx_mh_1208vioxx.html.
13. D. Wilson, "Merck to Pay $950 Million over Vioxx," *The New York Times*, 22 Nov. 2011; www.nytimes.com/2011/11/23/business/merck-agrees-to-pay-950-million-in-vioxx-case.html.
14. G. Rangwala, "[casi] Intelligence? The British Dossier on Iraq's Security Infrastructure," 5 Feb. 2003; www.casi.org.uk/discuss/2003/msg00457.html.
15. E. Gessiou et al., "Digging Up Social Structures from Documents on the Web," *Global Communications Conference*, IEEE, 2012, pp. 744–750.
16. "Metadata Ethics Opinions around the U.S.," American Bar Association, 2013; www.americanbar.org/groups/departments_offices/legal_technology_resources/resources/charts_fyis/metadatachart.html.

Simson L. Garfinkel is an associate professor at the Naval Postgraduate School. His research interests include digital forensics, usable security, data fusion, information policy, and terrorism. He holds six US patents for his computer-related research and has published dozens of research articles on security and digital forensics. Contact him at simsong@acm.org.