Date 7/27/2008

Hidden Data and Metadata in Adobe PDF Files: Publication Risks and Countermeasures

Enterprise Applications Division
of the
Systems and Network Analysis Center (SNAC)
Information Assurance Directorate

National Security Agency
9800 Savage Rd. STE 6704
Ft. Meade, MD 20755-6704
(410) 854-6191 commercial
(410) 854-6510 facsimile

What this Document Addresses

This paper describes procedures for sanitizing PDF documents for static publication. Sanitization for the purpose of this document means removing hidden data and dynamic content not intended for publication (for example, the username of the author or interim editing comments embedded in the file but not visible on any pages).

The types of PDF documents that are addressed by this document include those converted from source formats such as Microsoft Office, Adobe FrameMaker, and any other native application. PDF documents produced through unknown sources should also utilize these sanitization procedures for static output.

What This Document Does Not Address

This procedure does not apply to document types for which interactive content is intended for publication. Users who wish to employ interactive content in their PDF documents (such as fillable forms, 3D content, layers, form calculations, and embedded media) assume additional risk because these sanitization procedures do not permit the retention of these content types in the sanitized static output.

This paper does not address the issue of redaction, which is the complete removal of specific visible content within the source document (for example, the removal of an image or a name in the text of the document). That procedure is outlined at NSA's website: http://www.nsa.gov/ia (search for 'redaction').

Sources for Further Research

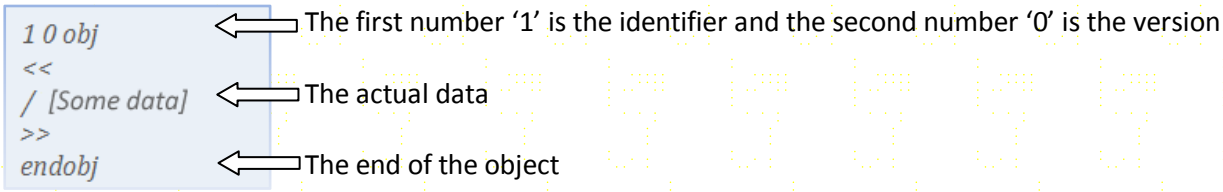
The PDF format has been approved as an ISO standard, ISO32000. Adobe also makes the PDF reference available on their website: http://www.adobe.com/devnet/pdf/pdf_reference.html .

Acknowledgements

The National Security Agency would like to thank Adobe for their technical contributions to this paper.

The Formatting of This Document

In order to better describe the contents of PDF files, examples of the binary contents are presented in some sections. To clearly identify these occurrences and make them easily distinguishable from the descriptive text, binary examples are presented in a blue textbox in the following form:



Executive Summary

The Portable Document Format (PDF) is pervasive and is used for publishing documents on the web, exchanging files between government entities and government contractors, and for interactive content such as forms and multimedia. Some of the reasons for its popularity include the wide availability of PDF authoring tools and freely available readers, cross-platform interoperability, and the ability to maintain the appearance of content across clients with varying hardware and software.

PDF has also been frequently used as a distribution format for files originally created in Microsoft Office because hidden data and metadata can be sanitized (or redacted) during the conversion process. Despite this common use of PDF documents, users who distribute these files often underestimate the possibility that they might contain hidden data or metadata. This document identifies the risks that can be associated with PDF documents and gives guidance that can help users reduce the unintentional release of sensitive information.

Table of Contents

Table of Figures

1 Introduction

The Portable Document Format (PDF) has become the de facto standard for sharing information electronically. The reasons for its success include the ability to retain the appearance of content across varying client types, a reduced threat of unintentional information leakage, and an increasing feature set that supports a broad range of user requirements. The increasing feature set and widespread use raise important questions about what types of hidden data these files may contain, how significant the risks are, and how the threat can be reduced.

When many people think of PDF files, they often think of Microsoft Office files that have been converted to this format. However, any application that can send output to a printer can interface with Acrobat's print driver to generate a PDF file. The potential for hidden data in these files varies significantly with the source application. In addition, new features such as forms may be created in Acrobat or using applications such as Adobe LiveCycle Designer that generate PDF output without requiring input from any external source documents. In those cases, additional hidden data may be transferred to the resulting PDF file.

There are many potential sources for data in a PDF document, but a user might not even know what the source was. As a result, if a user receives a PDF document and wishes to share it with a broader audience, it may be difficult to adequately determine whether sensitive data remains in the file unless they follow a careful sanitization procedure. New features and added functionality have also created additional opportunities for the unintentional introduction of sensitive data into PDF documents. This analysis defines the risks associated with PDF documents and outlines a procedure to reduce them.

2 Background

Adobe makes the PDF specification available on their website for developers in order to "foster the creation of an ecosystem around the PDF format".[1] The open availability of the specification enables a wide range of developers to create applications that can read and write PDF files. Variations (including errors) in specific implementations of these applications increase the complexity of evaluating the hidden data risk.

There are four major subsets for PDF, including PDF/X, PDF/A, PDF/E, and PDF/UA.[2] PDF/X is used mainly for graphics and printing. It includes enhanced color management and prohibits the use of active content that cannot be printed. PDF/A is used for archival purposes, and requires that major components such as fonts be embedded so that even if any dependent files are unavailable in the future, the file will still be displayed normally. PDF/E is intended for engineering with support for 3D content, rights management, and commenting features. Finally, PDF/UA files are designed for universal

accessibility, so that disabled users and those who use assistive devices can effectively view the content. This analysis focuses on the basic PDF format and does not provide specific guidance for the subsets.

The PDF format has been used as a 'safe' format for mass distribution to a wide audience, such as when files are posted to a public website. Guidance to NSA/IAD customers for redaction encourages the use of the PDF format.[3] In addition to the reduced hidden data threat, PDF can be optimized for web viewing and site visitors only need to have the free Acrobat Reader installed in order to view the file. This benefit has further enhanced the adoption rate of PDF files for both government and commercial users.

The inherent complexity of formats such as PDF that can contain a wide variety of content types increases the likelihood that sensitive data may be unintentionally retained in the file. Understanding the inherent risks requires at least a basic understanding of the PDF format. Even though the format is 'open', understanding the structure and the associated risks is not a trivial task. The sixth edition of the PDF specification (Version 1.7) is 1310 pages. The content of a PDF file is typically compressed and may include a variety of different compression types, so extraction of individual components for analysis requires a combination of tools and techniques.

3 PDF Structure

PDF is a binary format that is based loosely on the PostScript language, and adds structure and navigation capabilities. The detailed implementation of PDF is moderately complex, and there are a wide range of developers who have created applications that generate or read PDF. With each different implementation, the possibility exists that the specifications were not followed exactly. For robustness, the Acrobat Viewer can reconstruct some portions of a file that are damaged or malformed. This feature enables the widest possible range of users to access PDF files that come from varying sources, but it also means that a file that does not conform to the specification may still open in the reader and potentially be difficult to identify as corrupt.

The contents of a PDF file may generally be read by a third party application and can be decomposed to examine the contents. One exception is where encryption has been applied. Encryption affects how the file may be read, but does not necessarily apply to the entire file. For instance, a document may contain metadata. Metadata is typically stored uncompressed in a file, so it appears in readable form in unencrypted documents. In encrypted documents, the metadata stream is typically also encrypted. However, if Crypt filters are used, the metadata stream may override the document encryption and force its contents to remain unencrypted (using an Identity crypt filter). For this reason, encryption may have varying levels of impact on the readability of the contents of a PDF file and apply only to limited sections.

The overall structure of PDF is fairly simple. There is a header, a body, a cross-reference table, and finally the trailer. An example is shown in Figure 1. The header contains the version number of the file. The body contains the objects that comprise the file such as text, images, and fonts. The body may also contain object streams, which contain a sequence of PDF objects. The cross-reference table and/or cross reference streams can be thought of as an indexes, as they provide the locations of objects in the body. Finally the trailer provides the location of the cross-reference table and potentially other items as well. Additional data may follow the trailer, such as update sections, which contain changes made to the file in an incremental update that overrides previous content in the file.

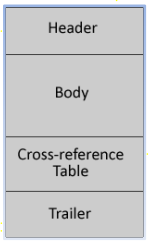


Figure 1 PDF Structure Overview

Header

A PDF file begins with the version number as the first entry in the header. The current version of Acrobat, version 8, can read files all the way back to version 1. The version appears in the first line of the file, and an example file that is version 1.5 would have an entry that appears in the form `%PDF-1.5`. For version 1.4 and later, this version entry may not be definitive, because an alternate version entry may exist in the catalog dictionary entry for 'Version'. A catalog dictionary is the root, or starting point, for accessing objects within a file. It describes how the hierarchy of the file is structured and includes information about how the file should be displayed in the reader. One of the scenarios where a difference may exist is when incremental updates have been added, and where the updates contain data that is a newer format than the base file.

While backward compatibility of new formats is not guaranteed, an older version of the readers and writers may still be able to read files created with newer versions of the file format. However, the newer features will simply be unavailable or hidden. The sanitization procedure that will be presented in this document recommends distributing the file in version 1.6 (Acrobat 7) because the active content that later versions facilitate will be removed during the regeneration process.

Body

The body contains objects of various types. Some of these types are simple, such as numbers, while others are complex. For example, subfiles with their own particular formats such as JPEG images, fonts, International Color Consortium (ICC) profiles, or PDF page descriptions are fairly complex. Objects may be stored as compressed streams, and content may be stored and compressed in a range of formats. Filters are used to describe how the data should be decoded. There are general filters and filters specifically intended for images. The general filters include FlateDecode, RunLengthDecode, LZWDecode, and ASCII85Decode, and ASCIIHexDecode. The image filters include DCTDecode, CCITTFaxDecode, JPXDecode, and JBIG2Decode. If encrypted data is contained in the file, the Crypt filter may also be present. The choice of format and level of compression vary with content type, but the range of options enables PDF files to be relatively small in relation to the size of the sum of their uncompressed parts. At the same time, this makes extracting and analyzing the contents trickier.