Global Information Assurance Certification Paper

Document Metadata, the Silent Killer…

GCIH Gold Certification

Author: Larry Pesce, larry@pauldotcom.com, lpesce@carene.org

Adviser: Rick Wanner

Accepted: March 27th 2008

Outline

Figures and Titles

1. Introduction

This paper will illustrate ways in which metadata stored in common types of documents can reveal secrets about an organization and how they can benefit an attacker. Throughout the course of this

paper we'll learn methods for auditing metadata exposure and some tips on assessing the risks associated with potential exposures. Additionally, we'll learn about some tools, their usage for auditing, discovery and proper sanitization. By the conclusion, the reader should have an understanding how metadata can assist an attacker as well as some process and policies to limit disclosure in the first place.

2. Background on Metadata

In a few short words, metadata is data that describes data. While that definition may not seem very interesting, the actual uses and applications are much more so. For purposes of this paper, we'll be examining that the metadata is describing the environment in which the document was created, or some properties of the document itself. Again, for purposes of this paper, we'll also be noting that the metadata is often "hidden" as it is not normally presented to the user.

In most applications, metadata is a fantastic tool for cataloging, indexing and searching quantities of documents. One certainly would expect to encounter document metadata in environments where large quantities of related, yet separate documents are utilized. One prime example would be that of a law firm, where legal documents authored by several people, for potentially hundreds of cases, could be indexed by metadata keywords for easier document retrieval, comparison, and determining possible precedence.

While this type of metadata most certainly has valid and useful purposes in business, or even at home, the actual contents it can reveal are often overlooked, especially when documents are placed onto the

Internet.

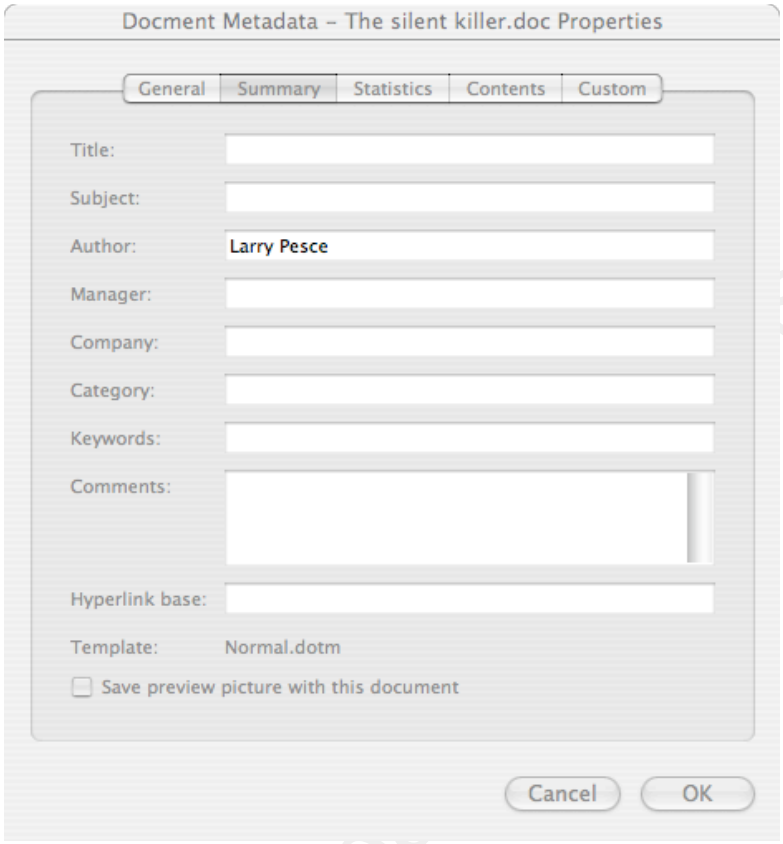
3. About Some Common File Types

Just about every electronic document that you can imagine contains some sort of metadata. We're going to focus the contents of this paper on some of the more common types, such as word processing documents and images. These types of documents can be found in just about every organization and home world wide, and they certainly can provide some very interesting information.

a. Microsoft Office

Most Microsoft Office documents are automatically populated with some form of metadata, some less obvious to the user than others. The first set that Office will include into a document can be found by accessing the document properties with File | Properties. Typically Office will pre-populate as much of this information as it can, most provided during the installation of the Office application. In the author's case, the only information that was pre-populated was the registered users name, as shown in Figure 1.

Figure 1: Minimal Pre-populated Office Document Properties



However, many users find this information helpful for tracking information about when or where it was created. Figures 2 through 4 show some metadata populated by the author, including some custom fields. These custom fields are user defined, but may be of the type that are useful for document and author tracking.

Figure 2: Document Properties Summary

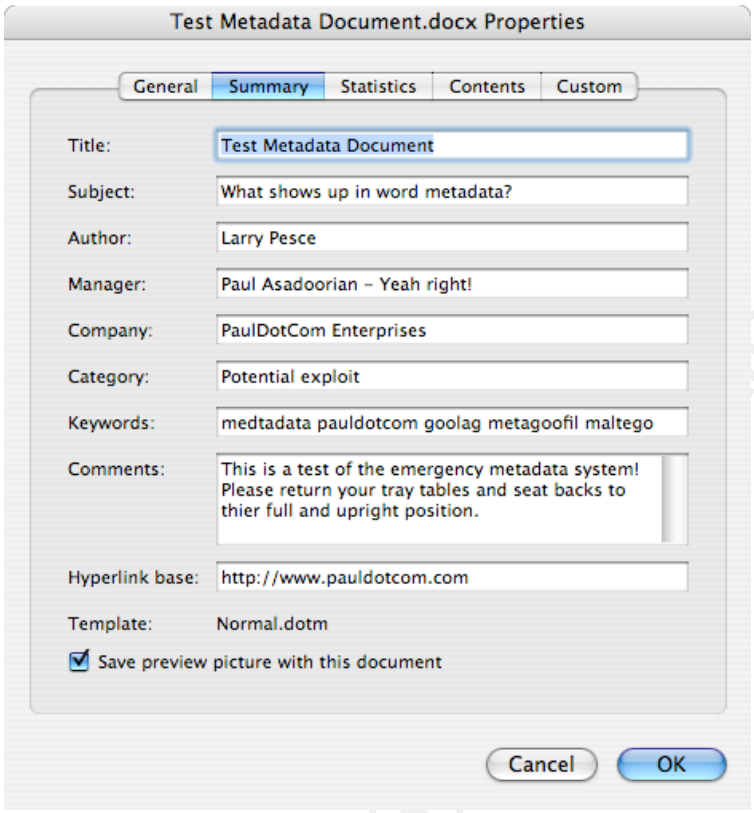


Figure 3: Document Properties Statistics

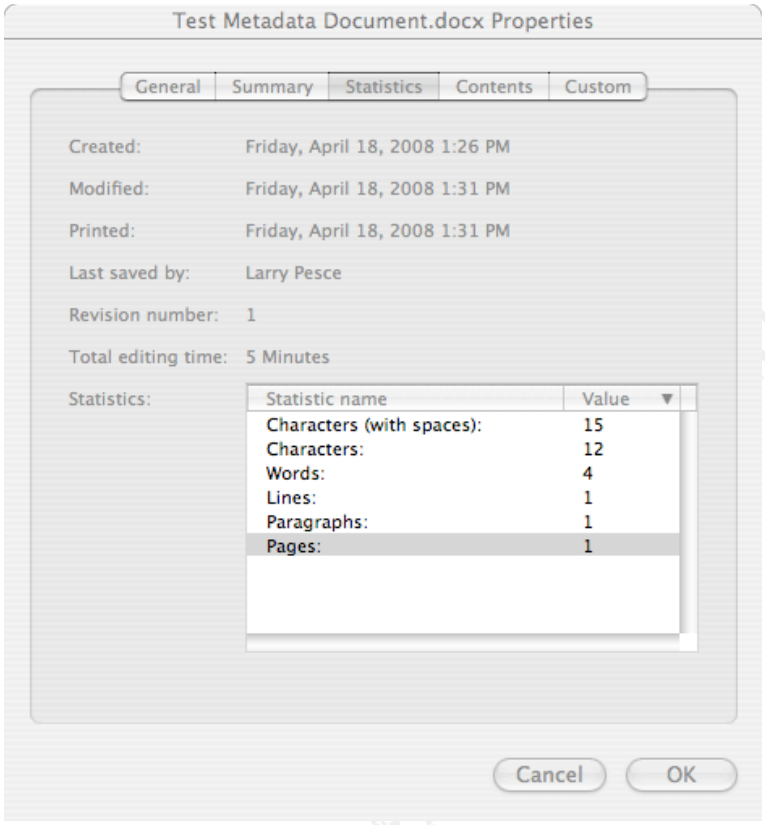
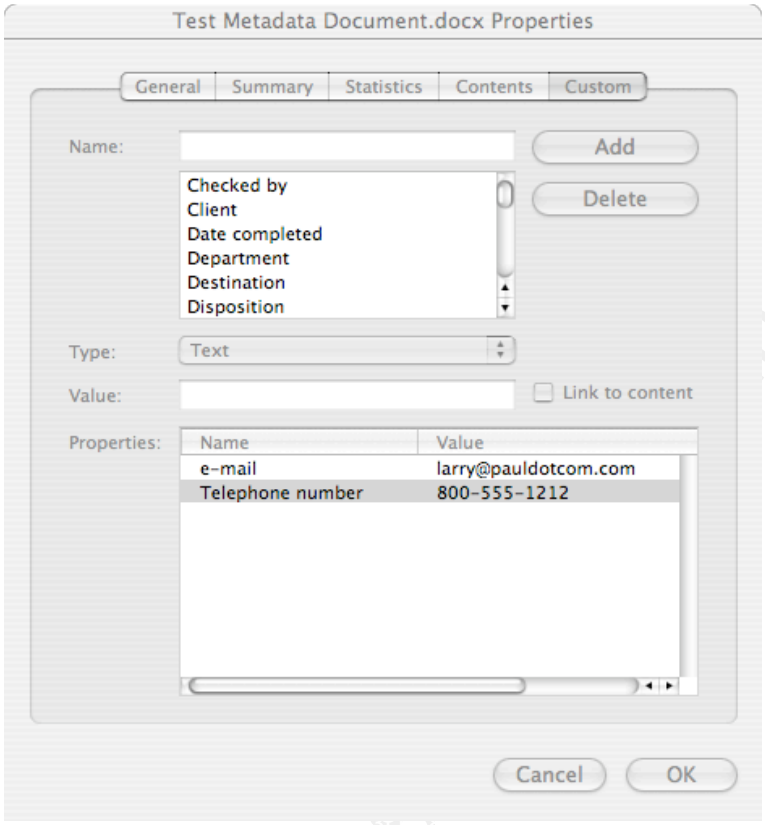


Figure 4: Document Properties Custom defined elements



In addition to these user editable and definable metadata objects, Office automatically includes a number of metadata objects that are not easily edited by the user. In many of these cases, the metadata is hidden from the user and exist mostly unknown to the document creator. As an example, we can use the Unix strings command on an Office document to reveal some of this information (which has been edited for space):

```
$ strings Test_Metadata_Document.doc
This is a test.
Test Metadata Document
What shows up in word metadata?
Larry Pesce
medtadata pauldotcom goolag metagoofil maltego
This is a test of the emergency metadata system!  Please return your
tray tables and seat backs to thier full and upright position.
Larry Pesce
Microsoft Word 12.0.1
```

```
Potential exploit
Paul Asadoorian - Yeah right!
PaulDotCom Enterprises
Test Metadata Document
Title
Telephone number
e-mail
800-555-1212
larry@pauldotcom.com
Microsoft Word 97-2004 Document
```

We can now notice some other pieces of important information, including the version of Word that was used, and some potential authors.

We should also note that the document creation dates and revision dates show up in the document properties, but are not editable by the user. Later on in this paper, we'll also indicate that there are some other interesting findings in the metadata of office documents, including MAC addresses, document file paths, usernames and text revisions left behind by the track changes feature.

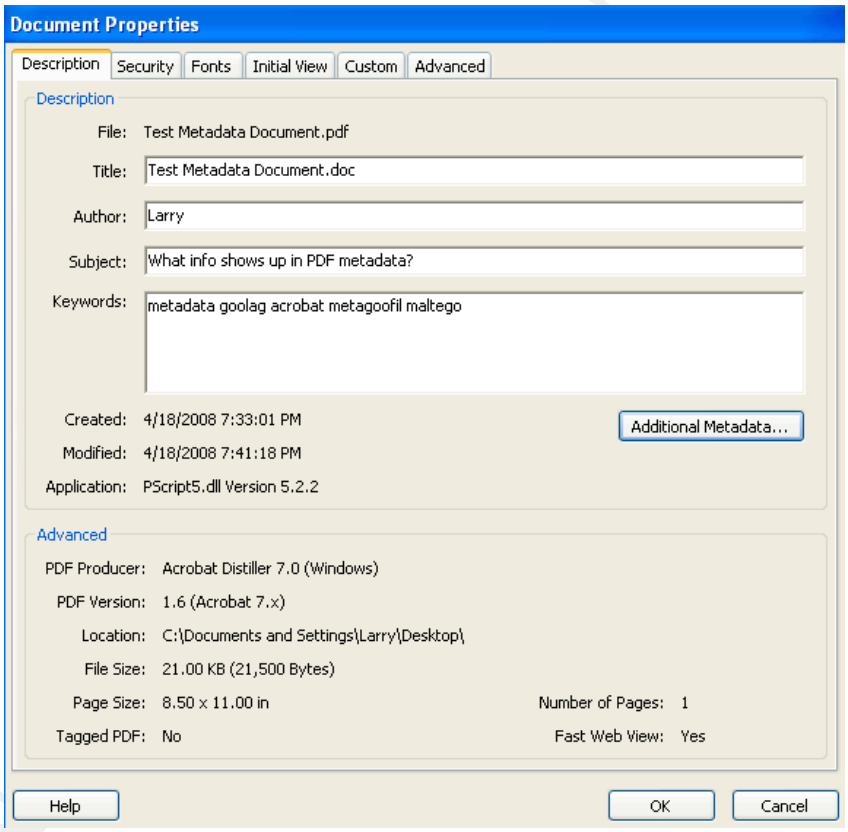
b. Portable Document Format (PDF)

PDF formatted documents have become the de-facto standard for transmitting documents across systems with disparate operating systems, while maintaining identical look and feel. This format is also restrictive in its editing capabilities so this format lends itself well to documentation, forms and other static documents.

In a similar fashion to Office document, Adobe's PDF creation tools automatically populate some metadata, of which some is less obvious to the user than others. These apparent, user defined metadata types that can be defined by Adobe's tools first can be found

by accessing the document properties with Adobe Acrobat Professional under File | Document Properties under the Description tab. Typically Adobe's tools will also pre-populate as much of this information as it can from the original document metadata. In the author's case, the information saved in the original Word document was populated in the PDF metadata as shown in Figure 5.

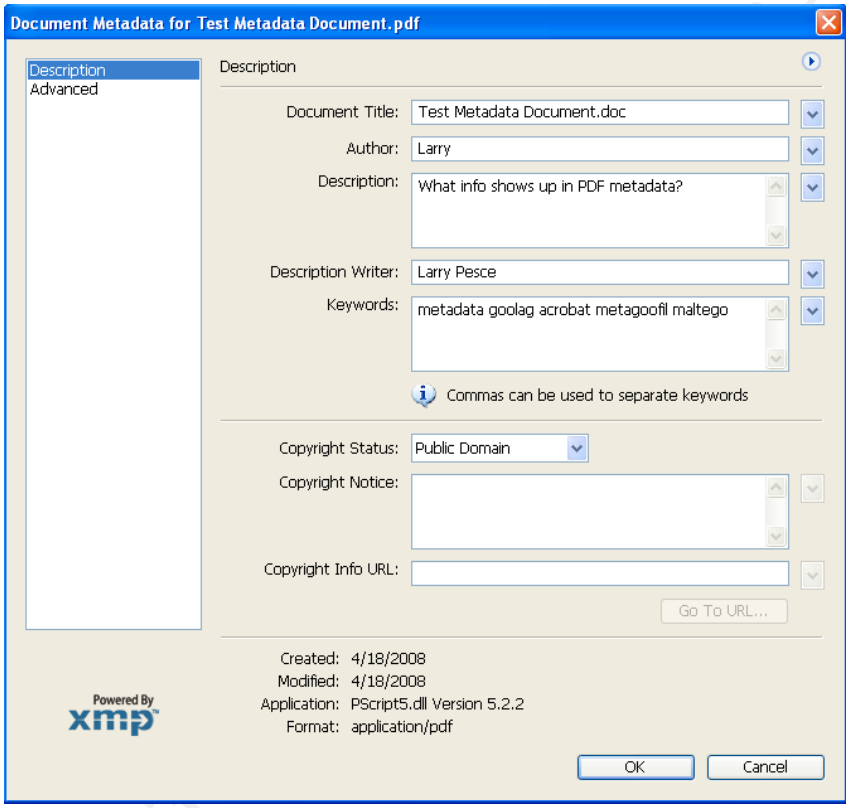
Figure 5: Pre-populated PDF Properties in Adobe Acrobat Professional



Again, many users find this information helpful for tracking information about when or where the document was created. These metadata types are also highly configurable by the user. These settings can be accessed in Adobe Acrobat Professional under File | Document

Properties under the Description tab, and by selecting Advanced metadata… as shown in Figure 6:

Figure 6: Advanced metadata in Adobe Acrobat Professional



In addition to these user editable and definable metadata objects, Adobe Acrobat Professional automatically includes a number of metadata objects that are not easily edited by the user. In many of these cases, the metadata is hidden from the user and exist mostly unknown to the document creator. As an example, we can use the Unix strings command on a PDF document to reveal some of this information (which has been edited for space):

```
$ strings Test Metadata.pdf
…
```

```
<pdf:Producer>Acrobat Distiller 7.0 (Windows)</pdf:Producer>
<pdf:Keywords>metadata goolag acrobat metagoofil maltego

<photoshop:CaptionWriter>Larry Pesce</photoshop:CaptionWriter>
<xap:CreatorTool>PScript5.dll Version 5.2.2</xap:CreatorTool>
<xap:ModifyDate>2008-04-18T19:35:38-04:00</xap:ModifyDate>
<xap:CreateDate>2008-04-18T19:33:01-04:00</xap:CreateDate>
<xap:MetadataDate>2008-04-18T19:35:38-04:00</xap:MetadataDate>
<rdf:li xml:lang="x-default">Test Metadata Document.doc</rdf:li>
<rdf:li xml:lang="x-default">What info shows up in PDF
metadata?</rdf:li>
/Author(Larry)/Creator(PScript5.dll Version 5.2.2)
<rdf:li>Larry</rdf:li>
<rdf:li>metadata goolag acrobat metagoofil maltego</rdf:li>
…
```

We can now notice some other pieces of important information, including the version of the creation DLL, and version, as well as the creation date, modification date, and Metadata creation date (in this example, the metadata was added after the original document conversion.

It should be noted that there are a multitude of PDF creation and conversion utilities for Windows, OSX and Linux. Of the limited number that the author has been able to test, most offer much of the same ability to either convert the existing metadata, or to add and modify with the conversion tool. As another example, the author converted a Word document to PDF with the built in converter in Mac Office. Again for this example we use the Unix strings command to reveal the metadata (which has been edited for space):

```
$ strings Test_Metadata_OSX_Office_Document.pdf
…
/Author (Larry Pesce) /Creator (Microsoft Word) /CreationDate
(D:20080418134209-04'00')
/ModDate (D:20080418134209-04'00') /Producer (Mac OS X 10.4.11 Quartz
PDFContext)
```

```
/Title (Microsoft Word - Test_Metadata_OSX_Office_Document.docx)
…
```

In the Mac Office example, we have been able to determine some additional information, including an application (Microsoft Word), the converter (Quartz PDFContext) and the host operating system and version (Mac OS X 10.4.11).

c. Joint Photographic Experts Group (JPEGs)

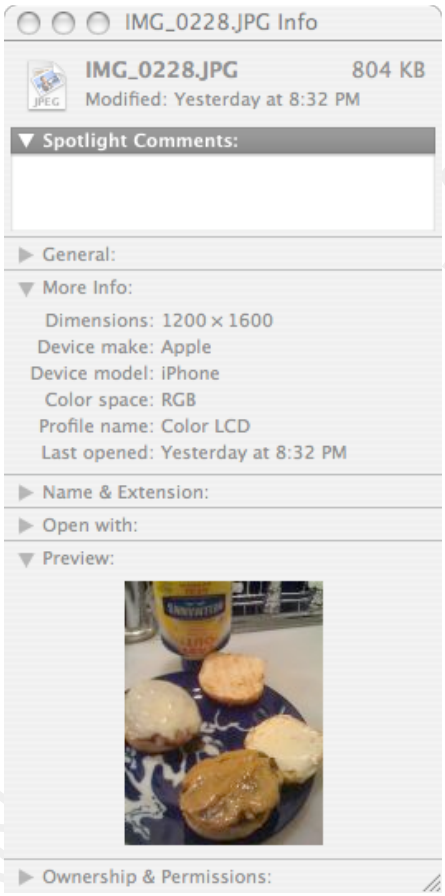
JPEGs have become extremely prevalent in today's digital lifestyle. They are created by just about every modern graphics program on the market, make up a large share of static image content on web pages, and are supported as output on all modern digital cameras in both professional and consumer grade model lines. It is no surprise that metadata in JPEGs can contain some very interesting information.

Unfortunately for purposes of this paper, analysis of any two output mechanisms, (whether it be graphics program or camera) would yield significant differences. Instead, we'll examine a few real world examples, as it is safe to say that most modern technologies support and retain JPEG metadata.

Metadata in JPEGs follows an open standard known as Exchangeable Image file Format (EXIF), which is an extension to the JPEG standard. Some common EXIF metadata includes the JPEG image creation data and time, camera settings, image description and even a thumbnail image. Often we will find that the utility, and even operating system that created the JPEG will be included. Included in the EXIF standard are hundreds of pre-defined tags for all types of information,

including the ability to add custom tags. We often find that examination of EXIF metadata yields a lot of chaff, with a little wheat. However, what wheat we do find will be valuable. As an example the image properties shown below under OS X by right clicking on the image file and selecting Get Info and expanding the More Info: section. This example as shown in Figure 7 indicates the image size, color profile, when the image was last opened, and the camera model (apple iPhone)

Figure 7: OS X display of limited EXIF Metadata



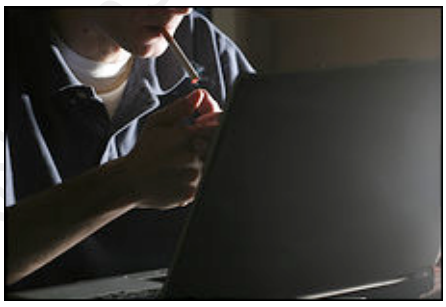
Yet another option to JPEG metadata is the Information Interchange Model by the International Press Telecommunications Council (IIM IPTC, or just IPTC as this metadata format is more

commonly referred as). In the case of IPTC metadata, the original use was for coordination ant proper crediting of photography across the major news wire services, such as the Associated Press.

In most cases common IPTC metadata types contains copyright and credit information of the photographer and news agency, output and processing information (such as camera or post processing software) as well as some descriptive text describing the contents of the image and location information (City, State and Country as opposed to Latitude and Longitude).

In 2006 an image was published of a hacker whom had admitted to computer crimes. The subject of the article, 0x80, was photographed in an anonymous fashion for the article by a photographer, who in accordance with the requirements for Associated Press photography included some interesting metadata. Below in Figure 8 is the photo of 0x80 that accompanied the article with an abbreviated Unix strings output revealing some EXIF IPTC photographer and location information:

Figure 8: AP photo of the hacker 0x80



```
$ strings Test_Metadata_OSX_Office_Document.pdf
…
Exif
SLUG:  mag/hacker  DATE:  12/20/2005 PHOTOGRAPHER:  Sarah L. Voisin/TWP
```

```
id#:  LOCATION:  Roland, OK
CAPTION:
PICTURED:
Canon
Canon EOS 20D
Adobe Photoshop CS2 Macintosh
2006:02:16 15:43:01
Sarah L. Voisin
0221
0100
2005:12:20 12:38:30
2005:12:20 12:38:30
0100
JFIF
…
```

As we can see in this particular instance we have used some limited tools to reveal the Photographer, Location (as documented by the photographer), Camera make and model, and some post processing software and the associated hardware platform.

While IPTC does provide some provisions for manual entry of location information, EXIF tags do provide location for Latitude and Longitude. Recent trends in photography automatically include location information automatically in the EXIF tags, known as geotagging (Dumell, 2006). In cameras that do not support this ability through built in hardware, additional modules are available. While this is add-on methodology is certainly known to the user, some other scenarios automatically include the location information without any intervention. As an example, the Apple iPhone (both revision 1 and the 3G version) photo taking application will gather location information by default, and with no interaction from the user. While the revision 1 iPhone does not contain a GPS unit in order to obtain location information, it will triangulate location based on known cell tower location under the 2.0

and later firmware. These features of either GPS location gathering or cell tower triangulation are also available in other cell phones including possibly the Nokia N95 as well.

As an example, the author was able to reveal EXIF metadata on an image taken with an iPhone revision 1 using cell tower triangulation. This photo was uploaded to Flickr, which analyzes and displays metadata information (Bjork and Sound, 2008). This EXIF display can be accessed while viewing the single image from the Flickr photo stream and selecting more properties from the right hand menu. Figure 9 shows the Flickr EXIF display of the author's image.

Figure 9: EXIF metadata display of location information on Flickr

Taken on
August 24, 2008 at 6.41pm EDT

Posted to Flickr
August 24, 2008 at 9.08pm EDT

Edit the photo dates

What is EXIF data?

Almost all new digital cameras save JPEG (jpg) files with EXIF (Exchangeable Image File) data. Camera settings and scene information are recorded by the camera into the image file. Examples of stored information are shutter speed, date and time, focal length, exposure compensation, metering pattern and if a flash was used.

Camera:	**Apple iPhone**
Aperture:	**f/2.8**
Orientation:	Horizontal (normal)
X-Resolution:	72 dpi
Y-Resolution:	72 dpi
Software:	QuickTime 7.5
Date and Time:	2008:08:24 20:44:32
Host Computer:	Mac OS X 10.4.9
YCbCr Positioning:	Centered
Date and Time (Original):	2008:08:24 18:41:43
Date and Time (Digitized):	2008:08:24 18:41:43
Color Space:	Uncalibrated
Latitude:	N 41° 52.1' 0"
Longitude:	W 71° 34.76' 0"
Compression:	JPEG

Return to the Mimi has weeds page.

While Flickr is a great tool for examining metadata in JPEG images, it is not terribly efficient. In later sections of this paper we'll examine some additional, more robust tools

d. Not Traditional Metadata, Yet Interesting

While the following two types of information are not usually classified as traditional metadata they do exhibit some of the same properties; they are typically not displayed to the user by default and provide valuable information about the content. Additionally this information is available on the Internet, can be used by an attacker or auditor to gather valuable information.

i. E-mail headers

In order for E-mail to function properly, each message relies on a series of routing information included as part of the message. This routing information is known as headers. These headers include information about the sender, recipient, server information (including IP addresses), and some relevant e-mail software, including the possible client application.

In most modern graphical e-mail clients, the Internet e-mal header information is masked from the end user. It is possible to reveal the headers by navigating some simple menu options in most clients. As an example, header information is available in Mail.app under OS X by selecting an e-mail message and selecting View | Message | Raw Source. Microsoft Outlook will reveal the headers by selecting a message, right clicking and selecting Options and viewing the Internet Headers box. Below shows a brief example of Mail.app's message

header output of a message addressed to the author.

```
Delivered-To: larry@pauldotcom.com
Received: by 10.65.40.11 with SMTP id s11cs103281qbj;
        Fri, 5 Sep 2008 06:46:28 -0700 (PDT)
Return-Path: <paul@pauldotcom.com>
Received: from johnnymo.paul.com ([74.14.86.36])
        by mx.google.com with ESMTPS id
p27sm274252ele.0.2008.09.05.06.46.15
        (version=TLSv1/SSLv3 cipher=RC4-MD5);
        Fri, 05 Sep 2008 06:46:20 -0700 (PDT)
Message-ID: <48C13821.3070804@pauldotcom.com>
Date: Fri, 05 Sep 2008 09:46:09 -0400
From: Paul Asadoorian <paul@pauldotcom.com>
User-Agent: Thunderbird 2.0.0.16 (Macintosh/20080707)
```

From this particular e-mail header, we are able to note e-mail server infrastructure, names, dates and e-mail client and associated OS platform of the author of the e-mail. It is important to note, that not only is this header information included with the individual e-mail messages, but that it may be disclosed in public mailing list postings, or in automatic Out Of Office replies. As an example the following sanitized OOO reply discussion was retrieved from an e-mail client subscribed to the Security Focus pen-test mailing list in which the user agent was detectable:

```
Received:  from lists.securityfocus.com (lists.securityfocus.com
[205.206.231.19])    by outgoing3.securityfocus.com (Postfix) with QMQP
id 6C53A237376; Sun, 14 Sep 2008 16:35:39 -0600 (MDT)
Content-Type: multipart/mixed;
    boundary="----_=_NextPart_001_01C916BA.781F8E05"
user-agent: Thunderbird 2.0.0.16 (Macintosh/20080707)
list-post: <mailto:pen-test@securityfocus.com>
list-id: <pen-test.list-id.securityfocus.com>
delivered-to: moderator for pen-test@securityfocus.com
mailing-list: contact pen-test-help@securityfocus.com; run by ezmlm
Content-class: urn:content-classes:message
Subject: EXAMPLE: Why OOO is *BAD* [WAS: Re: OOO FLAME]
Date: Sun, 14 Sep 2008 16:19:23 -0400
```

```
Message-ID: <48CD71CB.4070803@aset.com>
In-Reply-To: <00db01c9169c$53315120$f993f360$@com>
Thread-Topic: EXAMPLE: Why OOO is *BAD* [WAS: Re: OOO FLAME]
Thread-Index: AckWungd3zHVyhdvRauRbYpXN6N07Q==
From: "Tom Anderson" <neo@matrix.com>
Sender: <listbounce@securityfocus.com>
To: "Jack Sparrow" <captain@blackpearl.com>,
    pen-test@securityfocus.com
```

ii. GPG/PGP Key Trust Information

Certainly the outline of GPG/PGP operation and infrastructure is beyond the scope of this paper, it is important to understand at least one concept behind the encryption technology: Trust.

Trust with GPG/PGP is displayed by performing key signing; the act of having a third party validate that you are who you are, typically face to face, after verifying government issued IDs and verifying key checksums (Brennen, 2000). Then, the key signer applies their signature, or mark of trust, on the signee's GPG/PGP key, which is then published to public key servers. This act does actually require two individuals to have met in person, exchanged words, and interacted with each other, building a level of personal interaction. By providing personal exchanges, and exchanges of government issued identification, a certain level of trust between the two individuals has been established personally, as well as technologically.

When these additional key signatures are published to the public key servers, the additional trust information is included as well; this is how larger circles of trust can be established and verified. Of course, this key signing information is not reveled to the user during normal use of the GPG/PGP key or client.

As an example, we can use the web interface of MIT's public GPG/PGP key server at http://pgp.mit.edu to search for information by either e-mail address or by name. A search for the email address used in our e-mail header example returns a valid entry, as shown in Figure 10.

Figure 10: Search results at MIT's key server

Public Key Server -- Index ``paul@pauldotcom.com ''

```
Type bits /keyID    Date       User ID
pub  1024D/487FE094 2005/12/28 Paul Asadoorian <paul@pauldotcom.com>
```

By following the link indicated by the email address in this example, we can view who has signed the key for paul@pauldotcom.com for the key ID 487FE094, as shown in Figure 11 below.

Figure 11: Signers of paul@pauldotcom.com's GPG/PGP key

Public Key Server -- Verbose Index ``0x487F

```
Type bits /keyID    Date       User ID
pub  1024D/487FE094 2005/12/28 Paul Asadoorian <paul@pauldotcom.com>
sig         28988BF5           Roger Dingledine <arma@mit.edu>
sig         77FF506D           Matt Power <mhpower@mit.edu>
sig         4F5118FC           Matt Power <mhpower@mit.edu>
sig         699AADF3           R. Scott Buchanan <buchanan@cs.brandeis.edu>
sig         487FE094           Paul Asadoorian <paul@pauldotcom.com>
sig         487FE094           Paul Asadoorian <paul@pauldotcom.com>
```

From this output we can determine that paul@pauldotcom.com has had the GPG/PGP key with the ID 487FE094 signed by several individuals. We'll illustrate how this information is valuable to an attacker or auditor later in this paper.

4. Auditing Metadata and Assessing Risk

In this section we'll evaluate several methods and locations to audit for metadata, as well as offer some recommendations on

evaluating risk of the information exposures through metadata.

a. Common Places to Look for Metadata

While the places to begin looking for metadata are almost endless, we'll examine a few common places that pose some potentially high risk information disclosure.

i. Public Documents

Obviously, the Internet is a font of information that could turn up volumes about a possible victim. There are almost too many places to list to discover documents that contain valuable metadata. However, we should at least illustrate a few examples.

The first place that it makes to sense audit is the public facing website of the victim, or victim's employer. This is particularly effective to develop if you do not have a specific individual target in mind, as it will reveal potential individual targets. Manual enumeration of the website certainly works well, and you should be on the hunt for the type of documents we have used as examples; PDFs, Office documents and JPEGs.

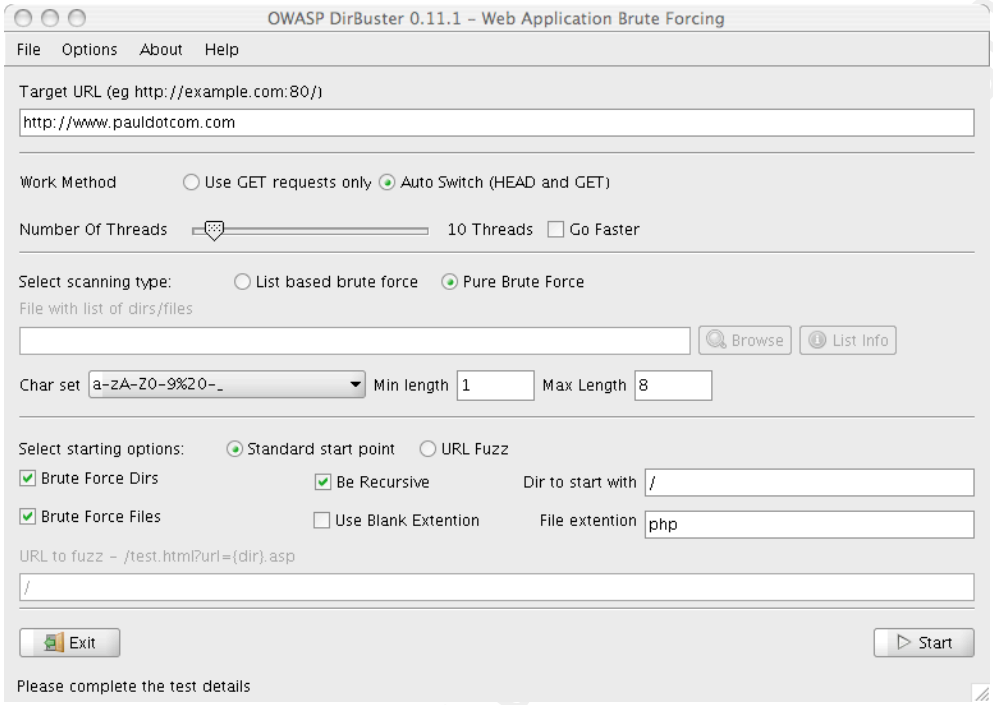
There are a few tools that might be helpful in examining websites remotely when you do not have administrative access to the web server host, in the same manner as an attacker. The first tool that is the easiest to utilize is the web hosts' robots.txt file (found at http://www.somesite.com/robots.txt). This file contains a list of files and directories that should not be indexed by search engines (Unknown, 2008); these locations often contain good information for metadata

analysis. Fortunately for the attacker or auditor the robots.txt is a double edged sword, as the file restricts what well behaved search engines should index, but also provides the same information to those who wish to utilize if for other purposes, such as for finding files that contain metadata that the organization did not want analyzed by search engines. As an example, below is the output for the robots.txt for sans.org (at http://www.sans.org/robots.txt). In this example the images directory may provide some interesting metadata: In the case of the sans.org website, access to the director is restricted, and/or directory listing is prohibited:

```
User-agent: *
Disallow: /images/
Disallow: /css
Disallow: /404.php
Disallow: /adminpage.php
Disallow: /registration/
Disallow: /jsf_detect.php
Disallow: /jsf_reg_detect.php
```

Another tool that may be used to find some interesting documents for metadata analysis is OWASP DirBuster (Sittinglittleduck, 2007). DirBuster connects to the specified website, and checks for the presence of subdirectories under the document root. An example screen shot is shown below in Figure 12.

Figure 12: DirBuster options screen



This checking can be performed using one or more included lists, or via brute force methods. Checking via the pre defined lists methods is infinitely faster, and the authors have already performed some validation of the lists (Sittinglittleduck , 2007). Pure brute force is certainly comprehensive, but can take quite a bit of time. During the author's last use of DirBuster to perform a pure brute force scan using the default options, DirBuster estimated that the scan would complete in 960,421,528 days (that's 2,629,490 years)!

Discovering personal websites of individual targets is an exercise left to the reader, however, Maltego featured in this paper can be a fantastic tool for that discovery process.

One other place to look for documents is the Secretary of State's office websites (or the office equivalent outside of the US). Often, these websites will contain PDF or Office documents intended to

become public records: articles of incorporation, annual reports, certain court and legal documents, and some legal applications that require public review. In some cases these documents are already indexed by Google, but in many cases they are not.

ii. Google

Billed as one of the most comprehensive, widely used search engines of the modern Internet (Sullivan, 2006), it is no surprise that Google is a valuable tool for gathering documents for metadata analysis on a particular target. Many of the automated tools utilize Google as a backend for information gathering.

Google is an extremely powerful tool, however it does have it's limitations; it will not locate files that have not been linked to by any other pages, so documents included on the web server may not be indexed, but may still be available; robots.txt and OWASP DirBuster may pick up these files and directories.

Because Google is so comprehensive, we can very easily create some search criteria while looking for documents that are just too onerous to possibly analyze, and many or the documents are out of scope. As an example searching for "pauldotcom" on Google returns nearly 28,000 results! Fortunately we can harness the power of Google and use several search operators to limit our scope of document search.

The first step to limiting our manual discovery of files through Google would be to restrict the domain in which we want to search. In our previous examples we've been able to determine that our example

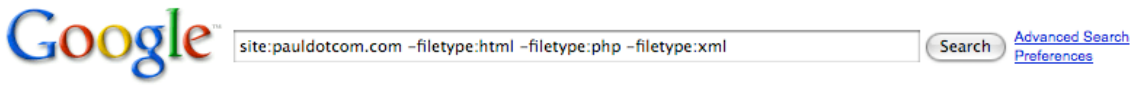
victim does a lot of work within the pauldotcom.com domain, so we'll use that as an example domain for our manual Google queries. In order to limit the domain search we will use the site: operator as shown in Figure 13:

Figure 13: Google site: operator

This search will now reveal all pages from the domain that Google knows about including any sub-domains (such as forums, www, and so on). This greatly reduces our search items to almost 2,200 results.

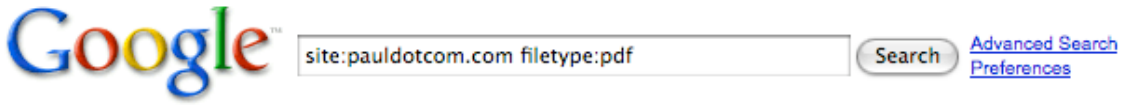
The reduced domain search still returns a bunch of stuff that we don't need for metadata analysis. There are two methods in with we can reduce our search; the exclusive method or the inclusive method. The exclusive method adds the –filetype tag (or several) as shown in Figure 14 to remove the resulting filetypes, which can drop us to about 360 results, many of which are not relevant.

Figure 14: Google -filetype: operator



With the inclusive search, we can pick specific file types that we wish to search for metadata on by using the plain filetype tag as shown in Figure 15. This method will reduce our results quickly, and will not introduce any extraneous irrelevant results.

Figure 15: Google filetype: operator



There are two issues however: With an inclusive search, we can only search for one file type at a time, and it does not play nicely for searches outside of the parent site (i.e., pauldotcom.com) and not any additional sub-domains (wiki, forum, etc.).

We can also harvest some information from Google on directories that allow directory indexing. Thee items will likely already be indexed by Google, but it can provide other useful information during information gathering, outside of the metadata scope. We can search Google in this fashion by adding the intitle: index.of search term to the site: directive as shown in Figure 16.

Figure 16: Google intitle: operator

It would also be bad form not to mention the powerful capabilities of Google's cache. Once Google indexes a site, it will maintain it's own cached copies of popular file types. This can be helpful to us as metadata analysts if the victim has removed the source file from the web server. Utilizing Google's cached document, we can still obtain it for analysis, even after the perceived threat has been removed from the victim's environment. It is certainly possible to remove the offending cached documents from Goggle, and we'll cover that later on in this paper.

iii. E-mail

As we have touched on earlier in this paper, we've discussed some of the type of information that can be found via e-mail based communications methods, be here we'll go into them in a little more detail. We'll cover direct e-mails (Out Of Office replies, bounces), as well as mailing list and newsgroup submissions.

E-mail traded between two individuals can reveal information about the client information as we have seen in the examples in the beginning of this paper. Of course this type of information can be obtained through the user agent (and other) fields of the e-mail header, the communication needs to be bi-directional in order for this to happen; the attacker needs to send an e-mail (likely from a dummy account), and the attacker has to respond, for the metadata to be returned user agent string to be sent back.

The establishment of the two-way communication can be forced through the inappropriate use of Out of Office (OOO) messages. When a victim sets an OOO message, often these responses leave the victim's organization. These can also be forced to be posted to mailing lists, if improperly configured on the victim's end. If part of a low traffic list, an attacker can possibly anonymously force the OOO to be sent to the list.

It is important to note, less technically savvy users may set OOO via rule, instead of via a wizard. When this happens, the client is left open, and the rules are processed on the client side, resulting in the typical user agent string inclusion in the mail header. However when, OOO messages are properly configured utilizing a wizard, the rules are

often created, and processed at the server side. This server side processing does not require any desktop client interaction, and as a result, we lose the user agent metadata. In most cases where the OOO is processed on the server side, the server will include some information about the server instead of some information about the client. This may be useful in determining attacks against mail server infrastructure and/or potential client software. As a specific example (individual mail servers and versions will vary), an email pulled from a public mailing list with a desktop client reveals the following sanitized e-mail headers:

```
Subject: [Email Tips] The Keymaker is out of the office.
Auto-Submitted: auto-generated
From: The Keymaker <TheKeymaker@matrix.com>
To: EmailTips@bogusmailinglist.org
Message-ID: <OF7E98F610.C0EF2284-ON852574E2.002DB5F6-
852574E2.002DB5F6@matrix.com>
Date: Tue, 14 Oct 2008 04:19:17 -0400
X-MIMETrack: Serialize by Router on D01ML076/01/M/IBM(Release
8.0.1|February 07, 2008) at 10/14/2008 04:19:18
```

From this message we are able to determine that the e-mail was auto-generated, as indicated by the Auto-Submitted header tag, as well as some interesting information in the X-MIMETrack header tag. A few Google searches on those unique characteristics reveal that the originating server was likely IBM Domino 8.0.1 with IBM Lotus Notes as a client. We're also able to tell about when the last release of the software was, as well as a date for the e-mail transmission, giving us insight into the possible patching practices for server side enterprise applications.

Newsgroup (Netnews, Usenet) submissions also feature very similar headers to that of e-mail. Newsgroup postings can take one of

two forms, either text or binary, and both can feature the same header information depending on the client. These headers as described in RFC 2045 (Freed and Borenstein, 1996) and RFC 2047 (Moore, 1996) are traditionally used to describe non-ASCII data included for binary newsgroup postings. In the author's experience, most modern news group posting clients do not differentiate between ASCII and non-ASCII postings, and include the appropriate header information, including user agent on both type of messages as shown in Figure 17.

Figure 17: Newsgroup header with defined newsreader

```
From: "M" <Mevi...@love.com>
Message-ID: <27g2l3.t58.17.1@news.alt.net>
X-Newsreader: Microsoft Outlook Express 6.00.2900.3028
X-RFC2646: Format=Flowed; Response
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3028
```

Again, with this type of information, we can utilize the X-Newsreader header to determine the client software in use on the victim's system. Couple that with the date of the posting, and we can make some continued assumptions about the timeliness of the client information.

5. Helpful Search and Audit tools

Now that we have established that there are several methods for obtaining valuable metadata, we need to discuss some helpful tools for finding our potential exposures on a more automated fashion than examining individual files one at a time.

a. Wget and EXIFtool

In order to analyze JPEG images from a website what we do not have direct access to (via the Internet as opposed to direct console or share access), one would normally consider utilizing a graphical browser. This process would send us down the road of navigating to the page, manually saving each jpg, and then analyzing each image. With a website of any considerable size, this task would end up being extremely time consuming. Utilizing the power of a few Unix based utilities wget and EXIFtool we can automate this process. Wget and EXIFtool are also available for windows and OS X, but we'll be covering the Unix/OS X variants here, but all command line options should be the same.

Wget is a command line utility for downloading web (and other) content, and storing it locally (Free Software Foundataion, 2008). The command line options for wget are tremendous, and we can utilize them to retrieve just what we want from a web server. We do need to be careful, as we can specify how many links deep we wish to follow within the website we wish to gather JPEGs from. This can quickly take us out of scope of the original website and while not illegal, it just adds extraneous information for us to analyze. We will only be using this method to retrieve JPEGs. At the Unix command prompt, in a temporary directory, we'll execute the following command:

```
$ wget -r -l1 --no-parent -A.jpg http://www.whitehouse.gov
```

This will execute wget to retrieve files recursively from the starting directory (-r), follow one depth of links contained on the page (-

l1, the one can be increased to follow links deeper), ignore the parent directory (--no-parent) in order to not traverse upwards in the event we specify a path after the domain in the URL, only store files ending in .jpg (-A.jpg) with the domain of http://www.pauldotcom.com. The results will be placed in a directory under the current path names after our domain (www.pauldotcom.com in this case), with a hierarchical directory structure (to the limit of our -l option) identical to that of the host website.

Now that we have retrieved the images, we also don't want to have to analyze each one individually. For that task, we can utilize EXIFtool, a perl front end to EXIF reading and modification libraries (Harvey, 2008). With EXIFtool we can retrieve EXIF metadata for all of the JPEGs downloaded by wget. To accomplish that task, we'll execute EXIFtool as follows:

```
$ exiftool -r -h -a -u -g1 * >output.html
```

This will execute EXIFtool to extract all EXIF metadata recursively in the current directory (-r), with all output including duplicates (-a), organizing by EXIF tag category (–g1), for all files (*, in this case only JPEGs as retrieved by wget), with HTML friendly formatting (-h), into a file named output.html in the current directory (>output.html).

The output file can be opened with your browser of choice, and information can be viewed for all analyzed images at once. The output is divided by image name, followed by the EXIF tags, as shown in Figure 18.

Figure 18: EXIFtool HTML output

ExifTool	
ExifTool Version Number	7.23
File	
File Name	IMG_0228.JPG
Directory	.
File Size	803 kB
File Modification Date/Time	2008:08:30 20:32:32
File Type	JPEG
MIME Type	image/jpeg
Exif Byte Order	Big-endian (Motorola, MM)
Comment	AppleMark.
Image Width	1200
Image Height	1600
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
Y Cb Cr Sub Sampling	YCbCr4:2:2 (2 1)
IFD0	
Make	Apple
Camera Model Name	iPhone
Orientation	Horizontal (normal)
Orientation	Horizontal (normal)
X Resolution	72
Y Resolution	72
Resolution Unit	inches
Software	QuickTime 7.5
Modify Date	2008:08:30 20:32:32
Host Computer	Mac OS X 10.4.9

EXIFtool also has a wealth of command line options, some of which we will utilize later in order to remove EXIF metadata. EXIFtool can analyze more than just JPEG images, and as an example when used to analyze a Word document (a PCI Self Assessment worksheet), EXIF tool was able to determine some interesting information as shown in Figure 19.

Figure 19: EXIFtool analysis of a Word document

ExifTool	
ExifTool Version Number	7.23
File	
File Name	saq_d_v1-1.doc
Directory	.
File Size	806 kB
File Modification Date/Time	2008:09:25 13:37:31
File Type	DOC
MIME Type	application/msword
FlashPix	
Code Page	1252
Title	Description
Subject	Self-Assessment Questionnaire
Author	JWallace
Template	Normal.dotm
Last Saved By	vmoschella
Revision Number	3
Software	Microsoft Office Word
Total Edit Time	0
Last Printed	2008:01:07 19:27:00
Create Date	2008:02:25 19:06:00
Modify Date	2008:02:25 19:06:00
Page Count	37
Word Count	10471
Char Count	59687
Security	0
Code Page	1252
Company	Visa
Lines	497
Paragraphs	140
Char Count With Spaces	70018
App Version	12 (0000)

These tests with wget and EXIFtool can be run as needed and at repeatable intervals. Additionally, EXIFtool can be executed on a directory structure of existing files if access can be had either directly at the console or via file share (SMB, NFS, etc.) of the web server you wish to audit

b. **Metagoofil**

Metagoofil is a python application for automating document searches via Google and metadata extraction (Martorella, 2008). Once the search, based on some command line options, is complete, Metagoofil will automatically analyze the documents for metadata, extracting key pieces of information, and creates an HTML based report. In addition to office documents, Metagoofil will also query for PDFs and OpenOffice documents, which can also be just as helpful in

obtaining appropriate metadata information.

Currently Metagoofil is a command line only tool that runs on Windows, Linux and OS X, however not without some issues. At the time of this writing, current versions of the dependent tools for Metagoofil (libextractor) appears to be subtly broken under only for Office documents under OSX, so it is recommended to use Metagoofil under Windows or Linux. Additionally, if you do use this tool often, it is highly recommended to update the tool from the author's website regularly; it utilizes output from Google via web page output. When Google updates search output, Metagoofil often does not know how to interpret the output and any resulting information or metadata gathering fails. Be aware that this tool may break Google's terms of service, and the tool's author does update the tool regularly to keep up with modified search engine output.

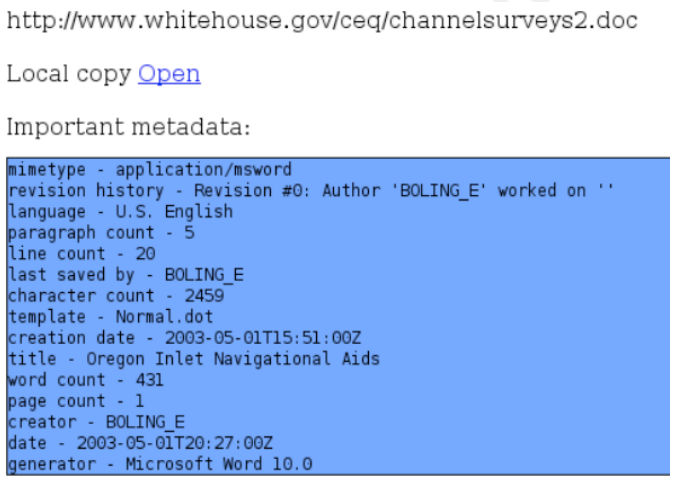
To begin using Metagoofil we start with what appears to be a complex command with several parameters:

```
python ./metagoofil.py –d whitehouse.gov –f all –l 1000 –o
whitehouse.gov-report.html –t whitehouse-temp
```

First we tell python to execute the application located in the current directory (python ./metagoofil.py), and then we pass it some information; the domain to search for documents (-d whitehouse.gov), type of supported documents we wish to analyze (-f all), limit on file count (-l 1000), an output file name for the HTML report (-o whitehouse.gov-report.html located in the current directory), and a temporary directory for file storage (-t whitehouse-temp located in the current directory).

Once complete we will be left with a report that breaks down the information in several different categories. The first section is a document by document breakdown of the interesting metadata contained in each, with a link to the original document analyzed (located in the temp directory). Each section will contain metadata about the document, that may include document creation tool and version, revision information, word count creation date and so on, depending on the document type. We can see an example of the individual document report in Figure 20, for a document analyzed from whitehouse.gov.

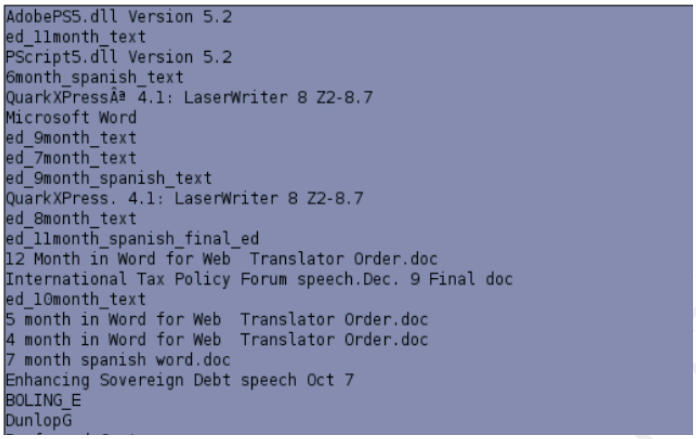
Figure 20: Metagoofil individual file report



Next in the Metagoofil report is a listing of extracted potential authors. This list of authors may include conversion tools, authoring tools, names and potential user account names. An example in Figure 21 shows some author results from whitehouse.gov.

Figure 21: Metagoofil author report

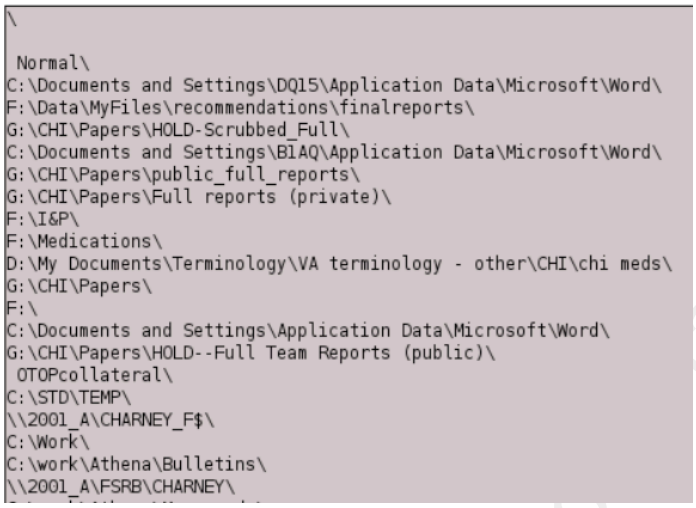
Total authors found (potential users):



Lastly, Metagoofil will tell us about all of the document paths that it was able to discover in our analyzed documents. These paths are typically indicative of where the documents are saved as a permanent location, and may be able to provide some starting information on where to begin searches for other sensitive information. It also can reveal other information about desktop policies (local disk storage), network drives (higher driver letters, and directory structure) and potential disk names (often included under OS X document paths). An example analysis from whitehouse.gov can be found in Figure 22.

Figure 22: Metagoofil document path report

Path Disclosure:



This analysis with Metagoofil can be run as needed and at repeatable intervals. Unfortunately at this time, Metagoofil will only perform the metadata analysis utilizing Google and a live host, and cannot perform analysis against a pre-acquired directory structure. The author of this paper has submitted a feature request for this capability to be added in future versions.

c. Maltego

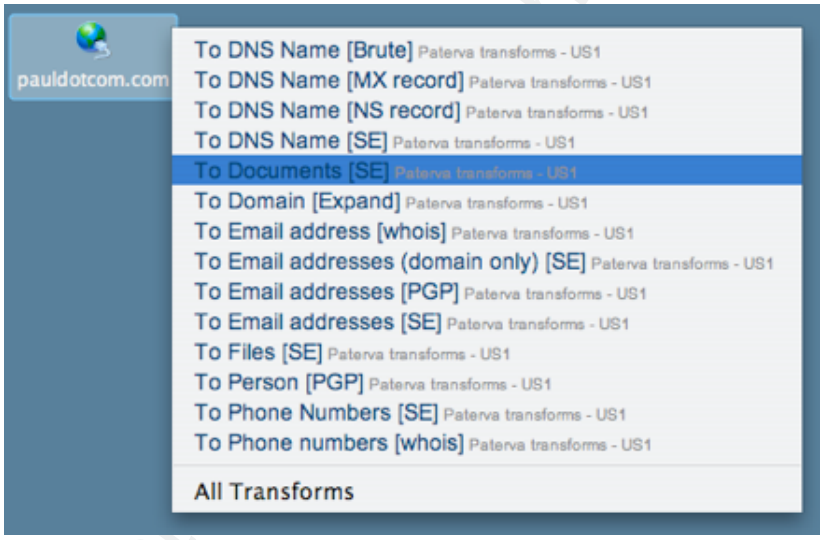
Maltego is the ultimate in information gathering tools. This tool features a GUI interface, running under Linux or Windows (and with some work, under OS X as well). It is completely extensible via a plugin architecture named transforms; each one performing a specified task to gather bits of information. Maltego is available in a free Community Edition (CE), and in a paid, unrestricted version (Temmingh, 2008).

Where Maltego's strengths can be found are in generalized information gathering, there are some abilities to decipher metadata son some common document types, including Office Documents and

PDFs. Additionally, Maltego is great for developing additional attack surface for an organization by utilizing the other transforms.

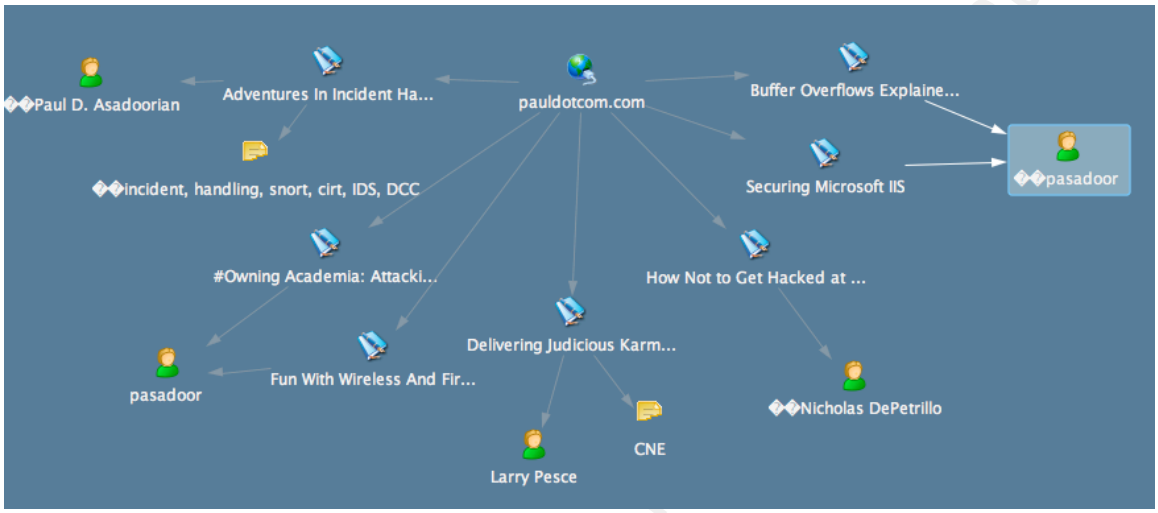
Once we've given Maltego a place to start, be it a website, email address, or person name, we can begin using the multitude of transforms to gather information. We can continue to determine more information on strictly documents by utilizing the To Documents transform, as shown in Figure 23, to gather all common document types associated with the current element.

Figure 23: Maltego To Documents Transform



Once we have gathered associated documents, we can use an additional transform to examine the metadata for interesting pieces, as shown in Figure 24, revealing a potential username for an associated document.

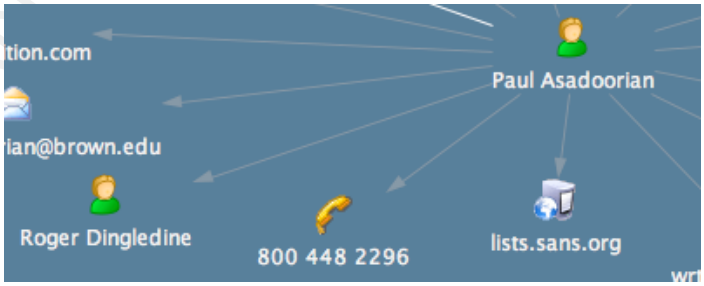
Figure 24: Maltego metadata display



While there may be more efficient ways of gathering this metadata, Maltego also helps to determine other relationships that may be valuable for attacks. As also seen in Figure 24 above, we can see that another user (Larry Pesce) associated with the target domain was able to reveal additional information about other associated organizations (CNE).

With Maltego, we are also able to determine some trust information based on PGP key signing. In Figure 25, we are able to see that there is a person relationship with the victim and Roger Dingledine, information we have manually gathered through PGP key trust information.

Figure 25: Person relationship information with Maltego



Maltego will also produce reports that contain the original look and feel of the gathered information, and the relationships that were found.

d. Automating manual searches

Through the use of our Unix/Linux command line tools, we now have the ability to utilize some repeatable automation for an ongoing audit process in or own organization. We can utilize sendmail and cron in combination with a simple shell script to perform the analysis and e-mail the results. Below is a sample script that can be used to obtain info with Metagoofil, wget and EXIFtool all in one shot and e-mail the results.

```
#! /bin/bash -x
#
# getmeta.sh - Metadata extractor shell script wrapper
#
# License and legal stuff:
#
# THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED
# WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO
# EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
# SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
# PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
# OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
# WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
# OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
# ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
#
# Original Author: Larry Pesce (larry@pauldotcom.com)
# Modifications: Paul Asadoorian (paul@pauldotcom.com)
#
#  - Revision History -
#
# .001 - Larry -Initial revision
# .01 - Paul - Fixed some bugs, changed directory structure
#


#
# Change these to reflect your environment
```

```
#

METAGOOFIL="./metagoofil.py"
EXIFTOOL="/usr/bin/exiftool"
DOCCOUNT=1000 #set to the number of each document type for Metagoofil to
downlaod

DOMAIN="www.site.edu"
OUTPUTDIR=./$DOMAIN
EMAIL="email@domain.com"
SMTP_RELAY=email-relay.domain.com

TEMPFILEMGF=$OUTPUTDIR/metagoofilresults
TEMPFILEWF=$OUTPUTDIR/exiftoolresults

#
# Make output directory
#

mkdir -p $OUTPUTDIR

#
# Execute Metagoofil, store all documents, and output results
#

$METAGOOFIL -d $DOMAIN -f all -l $DOCCOUNT -o
$OUTPUTDIR/metagoofil-report.html -t $OUTPUTDIR/docs > $TEMPFILEMGF

#
# Spider the site looking for jpg and images
#

wget -P$OUTPUTDIR/sitedump -r -l2 --no-parent -A.jpg http://$DOMAIN

#
# Execute exiftool on the files retrieved by wget
#

$EXIFTOOL -r -h -a -u -g1 $OUTPUTDIR/sitedump/* > $TEMPFILEWF

#
# E-mail results
#

cat $TEMPFILEMGF | sendmail -f$EMAIL -s$SMTP_RELAY $EMAIL
cat $TEMPFILEWF | sendmail -f$EMAIL -s$SMTP_RELAY $EMAIL
```

With this script, we can now edit our crontab and have this test repeated at regular intervals of our choosing. This can be used to

determine, at a minimum any state change in metadata by manual comparison, or if any newly developed controls are missed after an initial audit.

6. What Metadata Can Reveal

Now that we have reviewed some of the documents we'll be examining, we can now examine how an attacker can begin using this information to develop an attack surface, or evaluate risk. For the purposes of these examinations, let's refer back to the examples given in the first part, and assume that they were all created with equipment owned and published by paul@pauldotcom.com. We'll continue to introduce some additional examples throughout this paper. While some of the examples are real, some have been slightly adjusted in order to protect the innocent (or guilty!), but do reflect other real world scenarios that the author has encountered during the research for this paper.

a. What the Attacker/Auditor Sees

If we begin to look at some of the documents, we'll begin to notice several pieces of information across all of the document types that help us determine a potential attack, or exposure. The first that has some significant bearing is the concept of time. With all of the types of metadata that we've illustrated (with the exception of key trust information), have record of when the document was created or transmitted, and in many cases when the document was last modified. This will be valuable in determining the validity of an attack; is it reasonable to assume that the software or hardware used to create the

document still in use by the affected party? In some cases, we're able to determine the date of creation, and the last time it was modified, indicating the length of use of a particular version of software.

Another item to note across much of the types of metadata is the indication of the hardware platform that created it. We've illustrated that JPEG images reveal the camera type that took the picture (Canon 20D), including devices not classified as cameras such as smart phones (iPhone), as well as some hardware platforms that have had some involvement with image post processing (Macintosh). With PDF documents we have also been able to determine, through some export methods that some hardware platforms were utilized (Macintosh).

We've also been able to gather some information about the host operating system and version. By examining metadata we've been able to determine that these documents were created through some various methods as shown in Table 1.

Table 1: Document creation determinations

Document Type	Metadata Strings Discovered
Office documents output on OS X	Word 12.0.1 is OS X only
PDFs via Adobe	Acrobat Distiller 7.0 (Windows)
PDFs via Word	/Producer (Mac OS X 10.4.11 Quartz PDFContext)
JPEGs	Mac OS X 10.4.9 as a host computer
E-mail	Macintosh/20080707

Additionally, we've been able to gather a significant amount about software and version numbers installed on the client operating system through examination of metadata as shown in Table 2.

Table 2: Software version numbers

Software	Metadata Strings Discovered
Office	Microsoft Word 12.0.1
PDF creators	Acrobat Distiller 7.0 PScript5.dll Version 5.2.2
JPEG Authoring	Adobe Photoshop CS 2 Quicktime 7.5
E-mail Client	Thunderbird 2.0.0.16

Finally, we've also been able to determine some other interesting information as shown in Table 3.

Table 3: Other interesting metadata

Source	Metadata Determinations
JPEG Geotag and EXIF	Latitude: N 41° 52.1' 0" Longitude: W 71° 34.76' 0"
GPG/PGP	Key trust with Roger Dingledine
MAC Address	Wireless card, ability to determine possible driver (Ellch, 2006)
MAC Address	If Wireless card, likely laptop with portable data
GPS and EXIF	Hardware platform, likely smart phone of iPhone, Nokia N95 and others

b. Putting it All Together

Let's now begin to make some inferences as to what this information can lead us to for either an attack or more information. The pieces that we've found have been quite extensive, but how do they all fit together?

First off, we have been able to determine some location based information. If we were able to infer that this location is the home of the individual we wish to attack. In this day and age of a mobile work force, the changes are fairly good that the individual may take a laptop home with them with corporate data on in, or with VPN capabilities. By

performing reconnaissance against the non-corporate location, an attacker may be presented with an opportunity for theft of corporate equipment, in a more relaxed, less secure and unattended environment. We could certainly be making some educated guesses about the assignment of the laptop to the victim, if we are able to obtain MAC addresses from Office documents, as the first three octets could reveal the inclusion of a wireless card in the PC that created the specific Office document. Additionally, armed with the knowledge of the existence of wireless, and the manufacturer of the wireless card, in conjunction with the possible victims' location, we may be able to launch specific wireless attacks. These attacks could be tailored to very specific wireless chipsets and driver combinations (Ellch, 2006).

This reconnaissance of a home location could yield much more than just a laptop, as an attacker may discover valuable information located on a camera, or iPhone or other smart phone, which we've also been able to determine is in the possession of the victim. On the camera (Canon 20D), which is a "pro-sumer" grade camera, it may be possible to reveal documentation on corporate intellectual property (such as pre-release press photography). The iPhone or smart phone will likely contain internal address book information, as well as stored e-mail addresses. With the iPhone, it is also possible to establish a VPN connection, and cache the user credentials (Heary, 2008). These location based attacks could certainly reveal sensitive information if not secured.

We have also been able to determine several other pieces of information about the various Operating systems in use. We have been

able to discern that OS X version 10.4.11 is in use through examination of PDF documents, as well of some unknown version of Windows in order to create other PDFs. This allows us to note which types of remote network attacks may be possible, but may not give us enough information to be conclusive. In addition, we can derive some knowledge about possible Operating System patches, given that OS X appears to be at the latest version as of this writing. With that information could assume that the victim stays relatively up to date on OS patches, possibly ruling out remote network based attacks against the host OS.

On the application side, we have been able to note that there are some interesting pieces that have been revealed. We know that the victim does not appear to utilize a Microsoft e-mail client, but instead uses Thunderbird. We're also able to determine that under the victims' Windows installation, the version of Adobe Acrobat Professional is several versions out of date. This may indicate a reluctance, inability, or lax policy on application updates and upgrades within the corporate (or personal) computing environment. Armed with this knowledge, we can assume that and client software based attack may be significantly more successful than a remote, network based attack against the OS.

The use of GPG/PGP reveals to us the information that the victim does share some level of trust. We can then perform additional information gathering on the trusted key signers, and utilize these names to forge appropriate communications to the victim. In this example, a little research reveals that Roger Dingledine is the project leader and Director of the TOR project. Armed with this information, we

could spoof e-mails from Roger Dingledine, to deliver alleged information about the TOR project to the trusted victim with an embedded exploit of our choice; based on our educated guesses based on metadata information.

GPG/PGP may also indicate to us that this particular user is a "power user", as GPG/PGP is more esoteric for the average corporate computer user. However, the use of GPG, and the assumption that the victim is a power user is not mutually exclusive. In some cases, delivering GPG/PGP to the end user, as well as the publishing of keys may have been an activity performed by the corporate IT department in order for the victim to conduct specific job related activities. However, the act of having the GPG/PGP signed by individuals outside of the victim's organization would certainly indicate a more intimate knowledge of GPG/PGP, and would indicate a power user. This tells us as an attacker, that we need to be infinitely more cunning in delivering attacks, client side or otherwise.

Further analysis of Office documents often reveals information about the path in which the document was saved. This can provide valuable information to be used during an attack. For example, we can reveal a login ID if the document is saved to the Windows My Documents folder. When saved, the path information will be in the format of C:\Documents and Settings\<user id>\My Documents. Armed with this user information, we now know a valid account name that we can utilize for password guessing, share enumeration, and other authentication attacks.

The last piece of information that we are able to utilize from out

metadata is one of the most valuable in corroborating the validity of possible attacks. This corroboration can be determined from the creation and modification dates of each of the individual documents. We can use this information to determine whether or not it is likely that that application or operating system is still in use at the time of attack, by dating the documents. It does not make much sense for an attacker to deliver exploits against an application discovered through metadata analysis, if there is a high likelihood that the application has been upgraded. For example, if we were to examine the creation date of an Office document, and we can determine that it was created within the last few weeks, there is a high likelihood that there have been no significant changes to the Office suite (especially if there have been no patches released in that timeframe). In an example with the PDF created with Acrobat Professional under windows, we can see that the particular version and creation DLL is several versions out of date, but according to the metadata, was used very recently (at the time of this writing) to create and modify the PDF. With this information, we can make a reasonable assumption that an exploit for the older Acrobat Professional would likely be successful; the application should still be in use, and likely in the same version and patch state.

Through this same time based information, it may also be possible to gain some additional insight in to application patching operations by determining application version and last use date through metadata. Much like determining if the application could be used as a valid attack, recent uses of older software may also indicate a reluctance or inability to patch desktop applications.

When we've take all of this information and added it together, we know that we can find one or more specifically targeted and valid exploit against client side applications, or against specific wireless hardware types known to be in use. We also know some methods in which to deliver the attack in a client side manner, via e-mail or browser, by impersonating individuals known to the victim.

7. Interpreting Results for Risk

Now that we have some visibility that this information is out there and viewable by an attacker, we should make some assessments on the information as the perceived risk to the organization. Obviously determination on type and severity of risk will vary per organization and their mitigation strategies, so this section will be highly subjective based on the author's experience.

In some of the cases of metadata information disclosure, there is little to no practical method to remove this information after it has been disclosed, or prevent it from being disclosed. For example, it is likely that many e-mail and news group headers will not be able to be sanitized due to general operation, e-mail standards and closed source. While this information will reveal some information about possible infrastructure components, in many cases is a low risk situation. Information about desktop client on the other hand, may reveal some critical attack vectors, with unlikely methods for sanitization.

Certainly the real low hanging fruit (in this authors opinion) exists more through the items revealed from sources that are readily addressable, such as PDFs, Office documents and images. These

particular items are readily addressable with tools for sanitation, user education and policy. Additionally, with these items, they appear to reveal more detailed information in order to determine an attack vector; Software and versions, usernames, directory structure, hardware information, location and possible location.

If we have been able to make a determination on what we value as audit points for our metadata, we can examine each individual component for risk. As an example the reveal of potential usernames now has enabled an attacker half of what is needed to begin brute forcing other services. With the inclusion of software version, location and hardware information these data elements can significantly narrow the potential for a successful targeted attack. It is the author's opinion that these are the more critical data elements to evaluate.

The evaluations of the individual data elements should be compared against all of the corporate policies and programs related to any defense in depth techniques. If the organization feels that particular controls are effective for mitigating risk on any attack that may be able to utilize the extended information for determining a targeted attack, then they may be rated with a lower level of risk. However, it is the author's opinion that may organizations put too much faith in some of their defensive strategies; Signature based defenses are only as good as the signatures, how do you monitor and assure that there are no false negatives, and just because an attack does not exist today what about an attack tomorrow (or today that no one knows about!). It would be the author's recommendation to include this type of audit and remediation into the defense in depth strategy.

8. Remediation

Once we have evaluated the risk of our metadata exposure, we need to find a way to mitigate. This section will discuss some of the reasonable remediation methods for those items that can be controlled by the company; in many cases it would be in bad taste to ask an employee to restrict or modify their personal online habits.

a. Removing the Source

The obvious place to begin with mitigating metadata is in the places where cleanup is relatively easy; this also comes with the bonus of having the highest risk information disclosure. The first place to start is often the corporate controlled website containing all sorts of JPEGs, PDFs and Office documents. In an environment that we control as an organization, it is always easier to clean up.

The easiest way to remediate these metadata exposure risks is to simply delete them. Unfortunately, the reason that the documents are there in the first place is to fulfill an important purpose; for example, information and forms for customers, as well as images to spruce up the web site. Certainly, deleting the documents will work, but it is far form practical. It is the first step in a more comprehensive remediation effort. After the documents are deleted, they can be replaced with sanitized documents. In this section, we'll learn how to sanitize the documents, effectively remediating the risk.

b. Cleaning Up Google

As we have learned earlier in this paper, Google is a hugely useful

tool for both an attacker and an auditor for finding metadata information. By either using manual searches or some automated tools we are able to hunt down plenty of documents. With the search results, we are then able to directly access the website to obtain the document, and begin our metadata analysis. If we have remediated the files on our site and replaced them with sanitized documents, the documents that we will retrieve and analyze will not provide any valuable information. This is an ideal situation for risk remediation.

We do need to mind one of the features of Google; Google Cache. With Google Cache, when Google indexes the contents of the website, it maintains a separate copy of the document on Google's servers. So, while we may have cleaned up our local copies on our server, they do still exist with Google until they crawl the site next, in about 8 (or more) weeks. This may be too long if we have determined a high level of risk. We can ask Google to re-index (Unknown, 2007),, and remove links in five days by submitting the URL to http://www.google.com/intl/en/remove.html. Alternatively (Unknown, 2007), we can submit the site and URL for immediate re-crawling by visiting http://www.google.com/addurl.html.

This removal and resubmission process will re-process the entire website, including all HTML pages, Office Documents, PDFs and JPEGs as well. This solution will remediate most of the high risk documents that have been already cleaned, and will repopulate Google's cache with the new sanitized documents.

In cases of extreme urgency, Google can be asked to immediately remove listing from the search engine (Unknown, 2007), but submitting

the URL to http://services.google.com/urlconsole/controller. This will remove items based on an updated robots.txt file, however it will not automatically re-index the contents of the entire site. This will result in items important to the business or customers, non-indexed by Google. Certainly, a second modification of the robots.txt file to re-include the files (after appropriate sanitization), and a resubmission for Google to re-crawl would repopulate Google, making the documents available again for searching for website visitors.

c. But Wait, There's More...

With Google we've just hit the tip of the iceberg for online searches and potential cached documents. A prime example of that was related to the summer Olympic female gymnasts from China. It was alleged that several of the gymnasts were not of the appropriate age. An enterprising individual searched Google for one of the Olympians, He Hexin. The search turned up an interesting Excel spreadsheet listing the Olympians alleged real birth date, putting her Olympic participation in question. The spreadsheet was not retrieved from the official website, but from Google cache. Very shortly thereafter, the document was removed form Google cache. The individual was able to find the same document in the cache of yet a different search engine, Baidu (StrydeHax, 2008). This is a prime example of yet another searchable document cache outside of Google being utilized for information gathering.

While it would beyond the scope of this document to cover every possible repository in cache of search engines, the author would recommend examining all potential options for cache removal based on

the risk determined by the organization. In this day and age one may safely assume that once a document has been published to the Internet, it will forever live there albeit in harder to find forms.

9. Preventing Exposure

The easiest way to prevent the exposure, regardless of risk is to prevent (or limit) the exposure in the first place. In this section, we'll discuss some methods on preventing the exposures form a human and policy perspective as well as some technological solutions.

a. Organizational Policy and Procedure

The first thing that we'll want to accomplish it to remove all relevant metadata information before it gets posted to the Internet or leaves the company. Unfortunately there are few (if any) automated systems to address theses issues for multiple paths of publishing. This is where policy and procedure come in to play.

In many organizations, there is staff in specific roles when it comes to external, public communications. Often this role lies with in a corporate communication department and or marketing department. In smaller organizations, this rile may be just one of many duties performed by one person; other duties may include content development and technical support. In many cases, the staff members developing and publishing web (and other corporate content), are those that technically savvy, but not in the minute details. Often they are more concerned with the meat of the documents, rather than the metadata of the documents.

One way to help mitigate the risk of document metadata information disclosure is to maintain a separate document store for sanitized documents. This is useful for staff that need to provide documentation to customers that may not be available on the corporate website, and it also makes this public information accessible to all staff members. As a benefit of the separate store, the un-sanitized documents can be populated with metadata to be used with internal content management solutions.

With separate content stores for sanitized and un-sanitized information, it is helpful to develop some policies and procedures to help the data be migrated. The policies can include actions that must be accomplished; no distribution of sanitized documents, where to find appropriate sanitization procedures, as well as some information on enforcement. Procedures should indicate exact steps to sanitize documents, who should be performing sanitization, and who should be publishing documents.

For the publication of sanitized documents, it would be a good idea to have some segregation of duties between the content developers, document sanitizers and content publishers. In many cases, even in large organizations, the document sanitizer could be combined in to one of the other roles. Regardless of how the segregation of duties is split, staff can benefit from a technical can risk based education program for document publishing, as well as education on your particular policy and procedures.

Regardless of who publishes final content to the sanitized document store or website, either marketing, creative, or technical

types can benefit from well documented and tested policies and procedures in conjunction with a robust education plan.

b. Tools to Use to Clean Up

We've talked considerably about tools used for detecting metadata; we now need to discuss several tools that can be used to begin cleaning up the highest risk for exposure documents. While this list of tools is by no means comprehensive, we will reveal several tools that are free or inexpensive and can be deployed in an organization to limit metadata information exposure.

i. EXIFtool

One of the easiest steps to perform for metadata sanitization is to clean up JPEGs of EXIF metadata on a website. For this cleanup we will need either direct access to the JPEGs on the server itself, via file share or the most recommended way, is via a copy of an un-sanitized document repository before publishing. We'll be utilizing the same tool that we used to audit JPEG metadata to perform the cleanup, EXIFtool, which is available for any system that can support a perl interpreter.

In order to remove EXIF metadata form JPEGs, we need to execute EXIFtool with the following options as shown below:

```
$ exiftool –r -All= *
```

This command will remove all EXIF and IPTC metadata, by setting it to null (-All=) for all file types (*, but it can only perform the operation for compatible file types, including JPEGs), while performing the modification recursively from the current directory (-r).

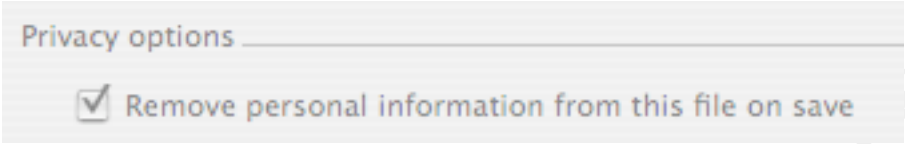
Results of this action can be reviewed and compared with our earlier audit command that was used in conjunction with wget. This method of removal will leave some document metadata behind in the document, but only that required for proper image rendering. Without the remaining metadata, the image can be considered corrupt rendering it unusable.

ii. Microsoft Office, Microsoft Document Cleaners and Third Party Tools

It makes sense to utilize the tool that you use to populate the document metadata to remove it as well. In this case Microsoft office products do a very good job or removing metadata. Unfortunately, most of the recommended methods for removing metadata for office documents vary, depending on Office version.

There are different guides for Office 97, Office 2000, Office 2002 Office 2003, and Office 2007 for manual removing metadata from documents, however automatic removal for all but Office 2007 is fairly straightforward. In your Office document, select Tools, Options, and on the Security tab make sure that Remove personal information from this file on save is checked, as shown in Figure 26. Once checked, Office will not save any personal information in documents. This setting is not a global change; it is a per document setting and is not a default Office setting.

Figure 26: Removing personal information in Office



Additionally, a plugin for Office 2002 and Office 2003 named "Remove Hidden Data" (Unknown, 2006) can remove document metadata from the command line. This tool, once installed can be found in C:\Program Files\Microsoft Office\Remove Hidden Data Tool\, and we can execute the following command to begin cleaning up Office metadata:

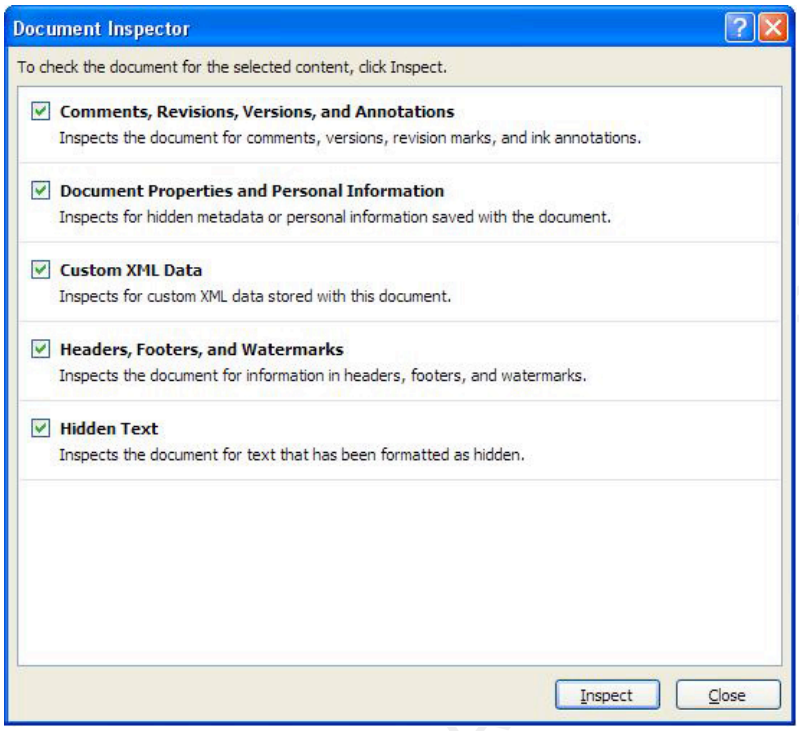
```
C:\Offrhd.exe C:\documents /R
```

This will remove metadata and personal information from all Office documents in the specified source directory (C:\documents), and perform the removal recursively (/R).

Office 2007 is a completely different animal. As of Office 2007, a new tool has been created and integrated directly into the Office suite named Document Inspector. Document Inspector will remove metadata from Office 2007 documents, and is backwards compatible with documents created with previous versions of Office.

We can use Document Inspector from within Office 2007 by selecting the Microsoft Office Button, Prepare, and then Inspect Document. We can then select the types of metadata we which to scan for and remove, as shown in Figure 27.

Figure 27: Document Inspector metadata selection



We will then select Inspect, and Remove All to remove all of the metadata that we have selected.

There are a number of additional tools from third parties, most of which do require a modest fee to purchase. In preparation for this paper, the author reviewed several that offered trial versions, and all offered similar functionality to the built in or free tools from Microsoft. With the third party tools, most did not support Office 2007 documents.

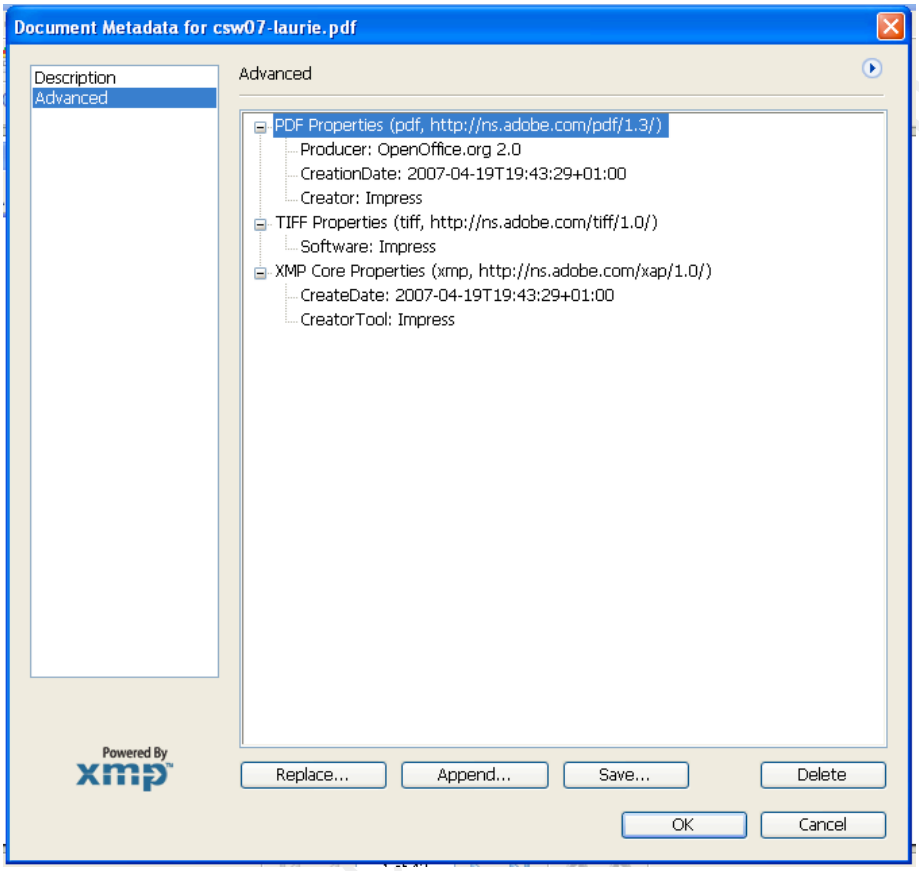
Regardless of removal method, the tools will still leave behind some information that is required for proper document utilization and may be required by the software. This can include software version that created the document in order to check for document compatibility.

iii. Adobe Acrobat & Third Party Tools

As with Office documents, one way to remove items in the PDF it to remove them with Acrobat at time of saving. This becomes a significant challenge when documents are converted to PDF format from a third party conversion tool or other authoring program. These third party converters often rely and populate the metadata carried over from the original authoring software. This can be removed by opening the final PDF document in Acrobat, with the exception of Acrobat Reader, assuming the document has not been protected.

In order to remove relevant metadata using Acrobat we need to select File, then Document Properties from the menu. In the new dialog box, we need to select the Description tag, then Additional Metadata. First, let's address the Advanced section as shown in Figure 28 below.

Figure 28: Acrobat Advanced Metadata deletion



By addressing the Advanced section first, we can delete one item and have it remove the rest of our Metadata items as a result, including those in the Description selection, as well as the properties screen. The complete removal can be accomplished by selecting the PDF Properties parent item and selecting Delete.

There are a number of additional tools from third parties, most of which do require a modest fee to purchase. In preparation for this paper, the author reviewed several that offered trial versions, and all offered similar functionality to Acrobat.

Again, much like Office document metadata removal, all of the tools will still leave behind some information that is required for proper document utilization and may be required by the software. This can

include software version that created the document in order to check for document compatibility.

10. Conclusions

After all is said and done, we can determine that document metadata has some valuable place in an information gathering and auditing program. This information can become valuable to an attacker, and most organizations don't realize that they have some form of exposure. Certainly these examples are only the tip of the iceberg for a determined attacker to formulate a detailed attack plan, based on document metadata alone. Even at that, it doesn't take much determination to gather some of this information. Information exposure via document metadata can be fun to audit and provide real risk for information exposure!

11. References

Bjork, G & Sound, H. (2008). EXIF Information. Retrieved November 15, 2008, from Digicamhelp: EXIF Information: http://www.digicamhelp.com/learn/glossary/exif.php

Brennen, V. Alex (2000, 10 01). The Keysigning Party HOWTO. Retrieved November 15, 2008, from CryptNET: Free Documentation Project Web site: http://www.cryptnet.net/fdp/crypto/keysigning_party/en/keysigning_party.html

Dumell, (2006, 08 21). Geotagging with Flickr. Retrieved November 15, 2008, from Geotagging with Flickr

| Life2go.net: http://life2go.net/geotagging_with_flickr

Ellch, J (2006). Fingerprinting 802.11 Devices. Monterey, CA: Naval Postgraduate School.

Freed, N. & Borenstein, N. (1996, 11). Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. Retrieved November 15, 2008, from Request for Comments: 2045: http://www.ietf.org/rfc/rfc2045.txt

Free Software Foundataion (2008, 02 07). Introduction to GNU Wget. Retrieved November 15, 2008, from GNU Wget: http://www.gnu.org/software/wget/

Harvey, P (2008). EXIFtool by Phil Harvey. Retrieved November 15, 2008, from EXIFTool by Phil Harvey: http://www.sno.phy.queensu.ca/~phil/exiftool/

Heary, J (2008, 07 30). How to build iPhone profiles for Cisco VPN. Retrieved November 15, 2008, from http://www.networkworld.com/community/node/30484

Martorella, C. (2008, 04 20). MetaGoofil - Metadata analyzer, information gathering tool. Retrieved November 15, 2008, from Edge-Security - Metagoofil - Metadata analyzer - Information Gathering: http://www.edge-security.com/metagoofil.php

Moore, K. (2008, 11). MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text. Retrieved November 15, 208 from Request for Comments: 2047: http://www.ietf.org/rfc/rfc2047.txt

Sittinglittleduck, (2007, 06 27). Category:OWASP DirBuster Project. Retrieved November 15, 2008, from Main Page - OWASP Web site: http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project

StrydeHax (2008, 08, 19). Hack the Olympics!. Retrieved November 15, 2008 from Stryde Hax: Hack the Olympics!: http://strydehax.blogspot.com/2008/08/hack-olympics.html

Sullivan, D. (2006, 08 21). comScore Media Metrix Search Engine Ratings - Search Engine Watch (SEW). Retrieved November 15, 2008, from Search Engine Marketing Tips & Search Engine News - Search Engine Watch (SEW) Web site: http://searchenginewatch.com/2156431

Temmingh, R. (2008). What is Maltego?. Retrieved November 15, 2008, from Maltego >> Home: http://www.paterva.com/maltego/

Unknown, (2008, 04 01). The Web Robots Pages. Retrieved November 15, 2008, from The Web Robots Pages Web site: http://www.robotstxt.org/robotstxt.html

Unknown, (2007). Stay Sharp: Google Hacking and Defense. Bethesda, MD: SANS.

Unknown, (2006, 11 23). How to minimize metadata in Word 2002. Retrieved November 15, 2008, from Microsoft Web site: http://support.microsoft.com/default.aspx?scid=kb;EN-US;290945