# Disclosing Private Information from Metadata, hidden info and lost data

Chema Alonso, Enrique Rando, Francisco Oca and Antonio Guzmán

**Abstract** — Documents contain metadata and hidden information that can be used to disclose private data and to fingerprint an organization and its network computers. This document shows what kinds of data can be found, how to extract them and proposes some solutions to the problem stated here.

**Index Terms** — Metadata, fingerprinting, security, privacy

## 1 INTRODUCTION

The collaborative work in on documents justifies the need of an extra information attached to the documents, in order to allow coherent and consistent results. In an environment where social networks make the sharing of resources such an important issue, it is necessary to store information about documents authors, the computers used to edit the documents, software versions, printers where they were printed, and so on. Then, if necessary, it will be possible, to prove the authorship of a concrete piece of information, to undo the last changes, to recover a previous version of a document or even to settle responsibilities when authorities want to investigate, for example, the management of the digital rights. The techniques used to attach this extra information to a document, without interfering with its content, are based on Metadata.

The concept of Metadata can be understood as information about the data. But it can also be understood as a structured description, optionally available to the general public, which helps to locate, identify, access and manage objects. Since Metadata are themselves data, it would be possible to define Metadata about Metadata too. This can be very useful, for example, when a given document has been the result of merging two or more previous documents.

The most frequent objective of Metadata is the optimization of Internet searches. The additional information provided by Metadata allows to perform more accurate searches and to simplify the development of filters. Therefore, Metadata emerge as a solution to human-computer communication, describing the content and structure of the data. Furthermore, Metadata facilitate further conversion to different data formats and variable data presentation according to the environment features.

Metadata are classified using two different criteria: content and variability. The first classification is the most widely used. You can easily distinguish among Metadata used to describe a resource and Metadata used to describe the content of the resource. It is possible to split these two groups once more, for example, to separate Metadata used to describe the meaning of the contents from those used to describe the structure of the content; or to separate Metadata used to describe the resource itself from those which describe the life cycle of the resource, and so on. In terms of variability, on the other hand, the Metadata can be mutable or immutable. Obviously, the immutable Metadata do not change, a typical example would be the name of a file.

The generation of Metadata can be manual or automatic. The manual process can be very laborious, depending on the format used for the Metadata and on their desired volume. In the automatic generation, software tools acquire the information they need without external help. However, only in few cases we can have a completely automatic Metadata generation, because some information is very difficult to extract with software tools. The most common techniques use a hybrid generation that starts with the resource generation itself.

If the information changes Metadata must change too. When the modifications are simple enough, they can be carried out automatically. But when the complexity increases, the modifications usually require the intervention of a person. In addition, the destruction of Metadata must be managed. In some cases, it is necessary to eliminate the Metadata along with their correspondent resources; in others, it is reasonable to preserve the Metadata after the resource destruction, for example, to monitor changes in a text document.

But the most critical situation is the destruction of Metadata when they are related to a final resource version intended for publication. The main contribution of this work is a research about what kind of information stored as Metadata in the public documents on the Internet is not destroyed and how this information can be used as a basis for fingerprinting techniques. We have selected two kinds of documents very usual on the web: Microsoft Office documents and OpenOffice documents.

## 2 METADATA AND HIDDEN INFORMATION IN OPENOFFICE DOCUMENTS

### 2.1 ODF FILES

ODF (Open Document Format) is the native file format used by OpenOffice, an open standard format, defined by OASIS and approved by ISO. In ODF, documents are stored as compressed ZIP archives containing a set of XML files with the document contents. If you use compression software to open an ODT document (text file created with OpenOffice Writer) you can find the following files:

- **meta.xml**: Metadata related to the document. This file is not encrypted even if the document is password protected.
- **settings.xml:** Information related to the document configuration and parameters.
- **content.xml**: File with the main content of the document, therefore, the text.


*Figure 1: ODT file contents*

Although OpenOffice version 1 uses different file extensions than OpenOffice 2, documents are stored in a similar way in both versions. Do not forget that ODF was built as an evolution of the file formats used in OpenOffice 1.

## 2.2 PERSONAL DATA

The first metadata generated using OpenOffice are created during the software installation and first execution. The software suite asks the user a set of personal data which, by default, will be attached to the documents created with this software.

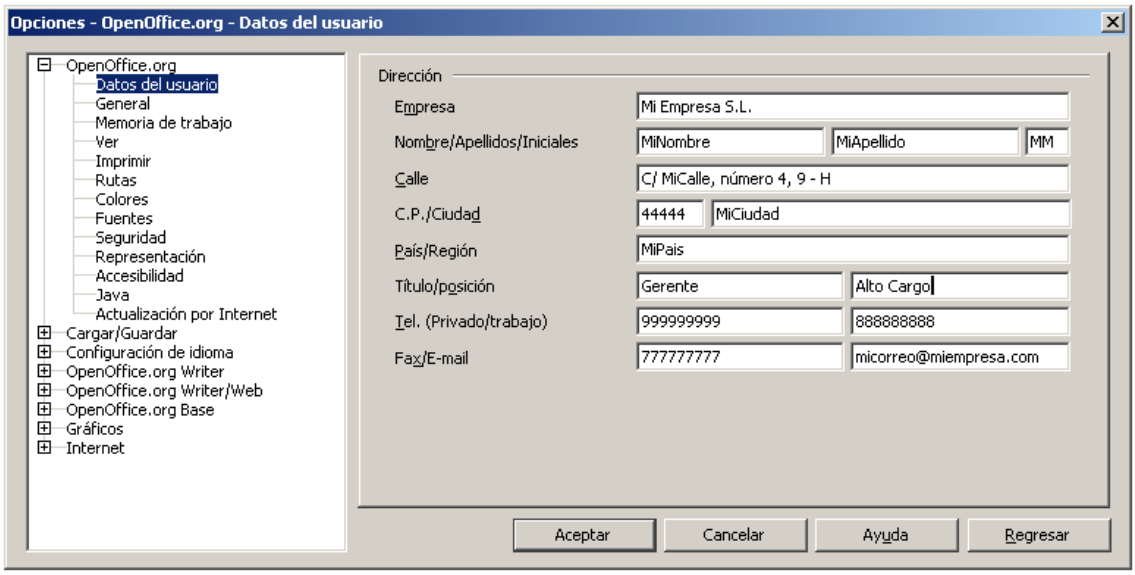


*Figure 2: User data modification*

Some of these data will be stored within the documents created by OpenOffice. If we create a new text document and afterwards check the contents of the generated meta.xml file, we will find the following information:

```
<?xml version="1.0" encoding="UTF-8" ?>
-<office:document-meta xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"
xmlns:ooo="http://openoffice.org/2004/office" office:version="1.0">
- <office:meta>
  <meta:generator>OpenOffice.org/2.3$Win32 OpenOffice.org_project/680m5$Build-
9221</meta:generator>
  <meta:initial-creator>MiNombre MiApellido</meta:initial-creator>
  <meta:creation-date>2008-08-11T11:33:23</meta:creation-date>
  <meta:editing-cycles>0</meta:editing-cycles>
  <meta:editing-duration>PT0S</meta:editing-duration>
  <meta:user-defined meta:name="Info 1" />
  <meta:user-defined meta:name="Info 2" />
  <meta:user-defined meta:name="Info 3" />
  <meta:user-defined meta:name="Info 4" />
  <meta:document-statistic meta:table-count="0" meta:image-count="0" meta:object-count="0"
meta:page-count="1" meta:paragraph-count="0" meta:word-count="0" meta:character-count="0" />
  </office:meta>
    </office:document-meta>
```

*Figure 3: meta.xml file*

We can find information about the OpenOffice version, the operating system, and, within personal data, about the user name. Perhaps we want to show this information or maybe not. A user or a company should decide about it before publishing the document on the Internet, before mailing it or before making it public by any other method.

### 2.3 PRINTERS

Among the data that can be potentially dangerous, because it reveals information about the company infrastructure, we have printer data. When you print a document with OpenOffice, and after it, you save the document; its settings.xml file stores information about the printer that has been used.

```
...
<config:config-item config:name="ClipAsCharacterAnchoredWriterFlyFrames"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="CurrentDatabaseDataSource" config:type="string" />
  <config:config-item config:name="DoNotCaptureDrawObjsOnPage"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="TableRowKeep" config:type="boolean">false</config:config-
item>
  <config:config-item config:name="PrinterName" config:type="string">EPSON Stylus DX4000
Series</config:config-item>
  <config:config-item config:name="PrintFaxName" config:type="string" />
  <config:config-item config:name="ConsiderTextWrapOnObjPos"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="UseOldPrinterMetrics"
config:type="boolean">false</config:config-item>
...
```

*Figure 4: printer information in settings.xml file*

This information may be important because it can denounce a forbidden action performed by a user or point directly to a specific user or machine uniquely. In terms of security this information could be even worse if the printer is shared on a server:

```
…
<config:config-item config:name="ClipAsCharacterAnchoredWriterFlyFrames"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="CurrentDatabaseDataSource" config:type="string" />
  <config:config-item config:name="DoNotCaptureDrawObjsOnPage"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="TableRowKeep" config:type="boolean">false</config:config-
item>
  <config:config-item config:name="PrinterName" config:type="string">\\servidor\HP 2000C
</config:config-item>
  <config:config-item config:name="PrintFaxName" config:type="string" />
  <config:config-item config:name="ConsiderTextWrapOnObjPos"
config:type="boolean">false</config:config-item>
  <config:config-item config:name="UseOldPrinterMetrics"
config:type="boolean">false</config:config-item>
   …
```

*Figure 5: Printer information described in UNC format in settings.xml file*

In this case, the printer appears in UNC format (Universal Naming Service), revealing both the server name and the correspondent resource. These data, for example, could be used by attackers to know the infrastructure of the internal network and to create a list of possible targets.

### 2.5 TEMPLATES

Templates are used to generate documents with predefined styles and formats. They are widely used because they allow using corporate documents and images comfortably. However, when a document is generated from a template, it stores references to the path location of the template in the meta.xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <office:document-meta xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"
xmlns:ooo="http://openoffice.org/2004/office" office:version="1.0">
- <office:meta>
  <meta:generator>OpenOffice.org/2.3$Win32 OpenOffice.org_project/680m5$Build-
9221</meta:generator>
```

```
  <dc:title>NuevaPlantilla</dc:title>
  <meta:initial-creator>MiNombre MiApellido</meta:initial-creator>
  <meta:creation-date>2008-08-12T10:02:14</meta:creation-date>
  <meta:editing-cycles>1</meta:editing-cycles>
  <meta:editing-duration>PT0S</meta:editing-duration>
  <meta:template xlink:type="simple" xlink:actuate="onRequest"
xlink:href="../../Datos%20de%20programa/OpenOffice.org2/user/template/NuevaPlantilla.ott" xlink:title="NuevaPlantilla" meta:date="2008-08-12T10:02:14" />
  <meta:user-defined meta:name="Info 1" />
  <meta:user-defined meta:name="Info 2" />
  <meta:user-defined meta:name="Info 3" />
  <meta:user-defined meta:name="Info 4" />
  <meta:document-statistic meta:table-count="0" meta:image-count="0" meta:object-count="0"
meta:page-count="1" meta:paragraph-count="1" meta:word-count="0" meta:character-count="9" />
  </office:meta>
</office:document-meta>
```

*Figure 6: Path to template in meta.xml file*

In the meta.xml file you can see the path to the template relative to the document location. This path may seem harmless and lacking the information that could put the system security at risk. However, if the document is stored in a folder located outside the user's profile, this path offers information about the user account.

```
...
  <meta:template xlink:type="simple" xlink:actuate="onRequest"
xlink:href="../Documents%20and%20Settings/UserAccount/Datos%20de%20programa/OpenOffice.org2/user/template/NuevaPlantilla.ott" xlink:title="NuevaPlantilla" meta:date="2008-08-12T10:02:14" />
  <meta:user-defined meta:name="Info 1" />
...
```

*Figure 7: Path to template in user's profile in meta.xml file*

In this case, the document has been stored in "C: \" and as a result, the path to the template reveals the folder that contains the user's profile in "C: \ Documents and Settings". The name of this folder is usually the name of the user account, in this example "UserAccount." It should be noted that, in certain cases, the name of this folder contains data about the domain to which the user belongs. This information is usually included in the name of the folder with the user's profile with the structure "UserAccount.DomainName" offering critical information to a potential attacker.

Similarly, the document could have been saved on another drive different from the template, obtaining in this case a complete path to identify the disk drive:

```
…
  <meta:template xlink:type="simple" xlink:actuate="onRequest"
xlink:href="/C:/Documents%20and%20Settings/papa/Datos%20de%20programa/OpenOffice.org2/user/template/NuevaPlantilla.ott" xlink:title="NuevaPlantilla" meta:date="2008-08-12T10:02:14" />
  <meta:user-defined meta:name="Info 1" />
…
```

*Figure 8: Full path to template in meta.xml file*

Examples shown above have all been performed on Windows machines, but the results do not differ much in Linux machines. In this case, paths to templates can contain information about user's $HOME Path:

```
…
  <meta:template xlink:type="simple" xlink:actuate="onRequest" xlink:role="template"
xlink:href="/home/pruebas/.openoffice.org2/user/template/PlantillaNueva.ott"
xlink:title="NuevaPlantilla" meta:date="2008-06-30T09:13:20" />
  <meta:user-defined meta:name="Info 1" />
…
```

*Figure 9: Full path to template related to $HOME in meta.xml file*

Logically, if the template is located on a network server, the information in UNC format shows the server's name and the shared resource, again allowing a potential attacker to reconstruct the network structure of the organization.

### 2.6 EMBEDDED AND LINKED DOCUMENTS

One of the options provided by almost all of the current office software is linking and embedding documents. In the case of linking files, there is a reference to the linked document in the main document, in the form of a relative path, when it is possible, and as an absolute path when there is no other alternative. If the document is linked on the same computer where the main document is, the result will be, in general, similar the results shown in the last section. Therefore, a potential attacker could disclose sensitive information about user accounts or file locations.

If the linked document is stored on another computer, the information disclosed to the attacker is very useful again:

```
…
<text:p text:style-name="Standard">
<draw:frame draw:style-name="fr1" draw:name="gráficos1" text:anchor-type="paragraph"
svg:width="16.999cm" svg:height="6.369cm" draw:z-index="0">
  <draw:image xlink:href="//desktop/confidenciales/Dibujo.bmp" xlink:type="simple"
xlink:show="embed" xlink:actuate="onLoad" draw:filter-name="<Todos los formatos>" />
  </draw:frame>
  </text:p>
…
```

*Figure 10: Linked document*

When the file is embedded in the document, not linked, there are not routes implied in the process, but we have to face new potential problems of leakage of information, because they may contain metadata and hidden information.

Suppose that we embed a JPG image (with its metadata in EXIF format) in an ODF document. In the example, one of such EXIF metadata is a miniature that looks different from the embedded image, thus showing that the image has been manipulated.

All the embedded files are included in the master document, so opening the ODT file with a decompressor, you can see that there is a folder called Pictures, and inside it is the embedded image, but under another name.

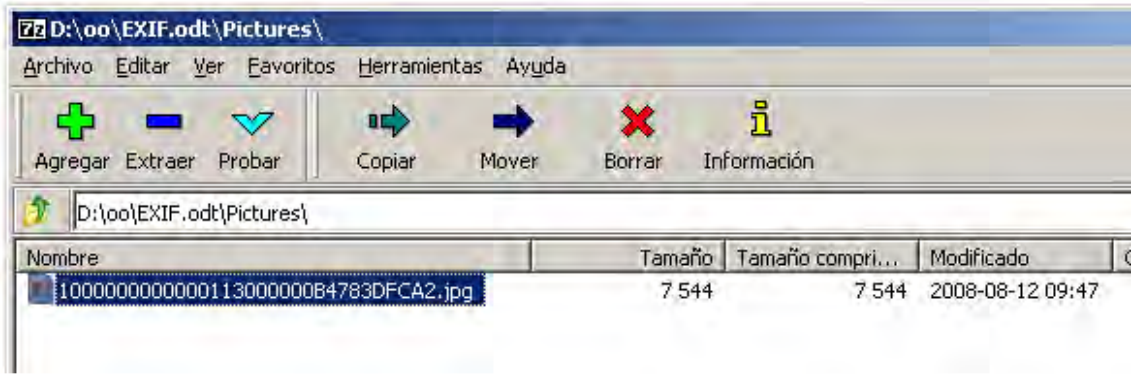


*Figure 11: Embedded image in Pictures folder*

If this image is extracted and analyzed, it has all the original image's metadata attached and, of course, the thumbnail that shows its original state. It´s possible to use any EXIF reader tool to analyze the thumbnail attached to the pictured to prove this as can be seen in Figure 12.



Figure 6 *EXIF Metadata from an embedded image*

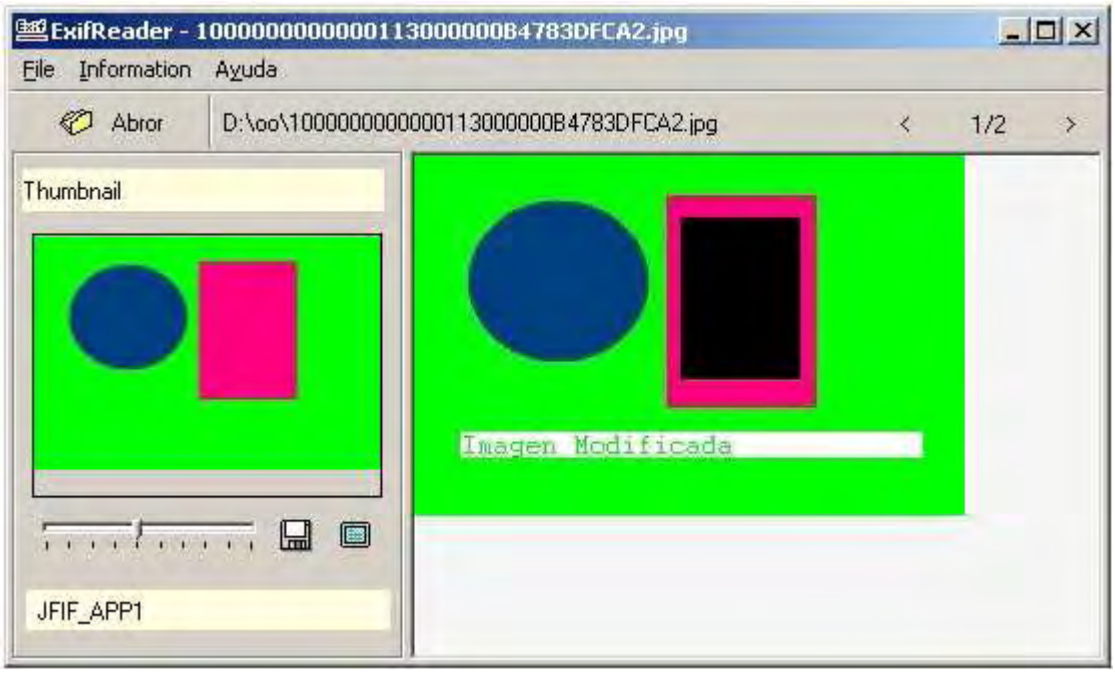


*Figure 12: Original thumbnail discover image's manipulations*

### 2.7 MODIFICATIONS

One of the features offered by OpenOffice Writer is to track changes in documents. This is useful when a document is being developed by multiple users or when you want to log all the modifications. The submenu "modification" of the Edit menu can activate this feature, as well as make visible or hide the changes.

A user can work on a document with this option activated, even by negligence, so that if the document is published without eliminating its history, anyone can guess if something has been removed or added, and by whom, and when these changes were made.

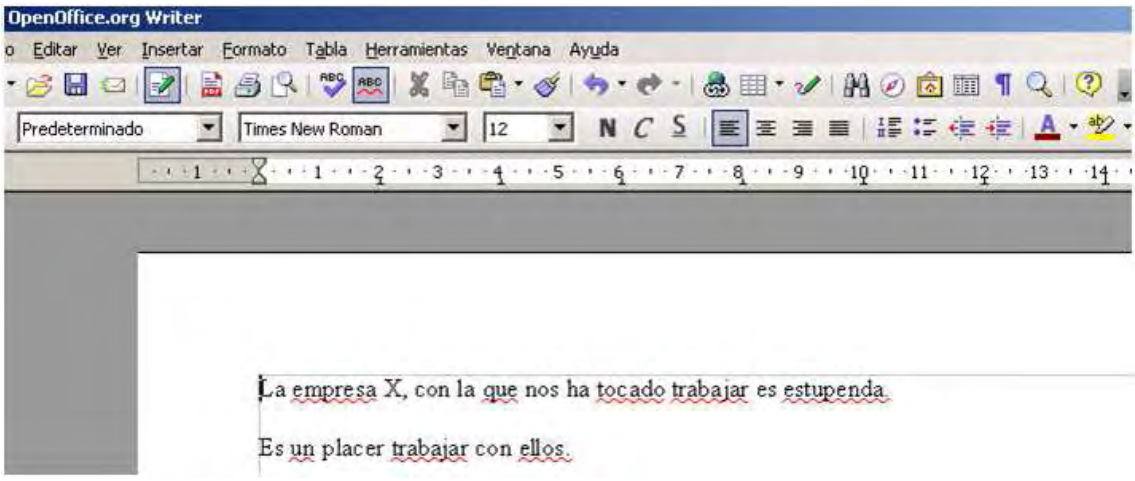


*Figure 13: Changes aren´t displayed*

Figure 14 shows that if you left the cursor on a change in the document for a few moments, there is a message indicating who made it and when.

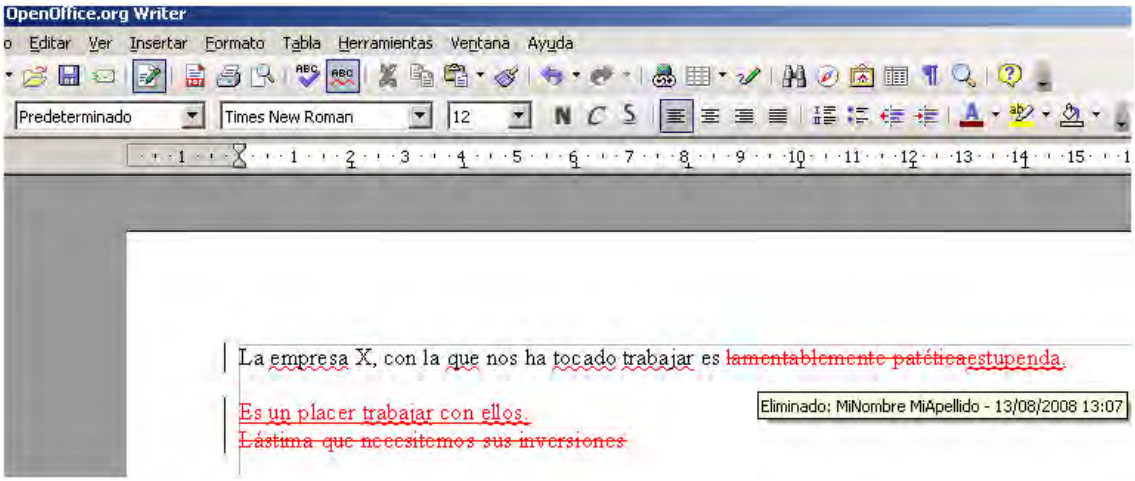


*Figure 14: Changes are displayed*

All of this information about the change history is stored in the file content.xml:

```
…
<text:tracked-changes>
 <text:changed-region text:id="ct110732472">
 <text:deletion>
 <office:change-info>
  <dc:creator>MiNombre MiApellido</dc:creator>
  <dc:date>2008-08-13T13:07:00</dc:date>
  </office:change-info>
  <text:p text:style-name="Standard">lamentablemente patética</text:p>
  </text:deletion>
  </text:changed-region>
…
```

### 2.8 HIDDEN PARAGRAPHS

Another option offered by OpenOffice is to hide text or paragraphs. This functionality allows working on a document in a display with hidden paragraphs, ready to print, and in another display, with all the paragraphs visible, for example, with the information for editing the document. This feature is activated including a special field in the paragraph:

We can turn on or turn off the display of hidden text using the corresponding menu item "View". So, if we have a document with hidden paragraphs, but we do not have the option of seeing them, we are working with a display that does not show all the information that the document has.

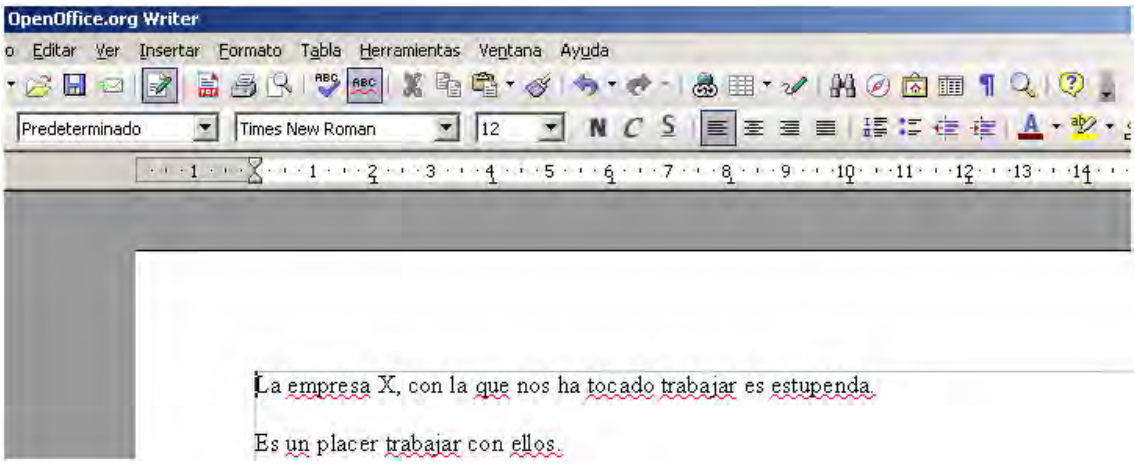


*Figure 15 Document with hidden paragraphs*

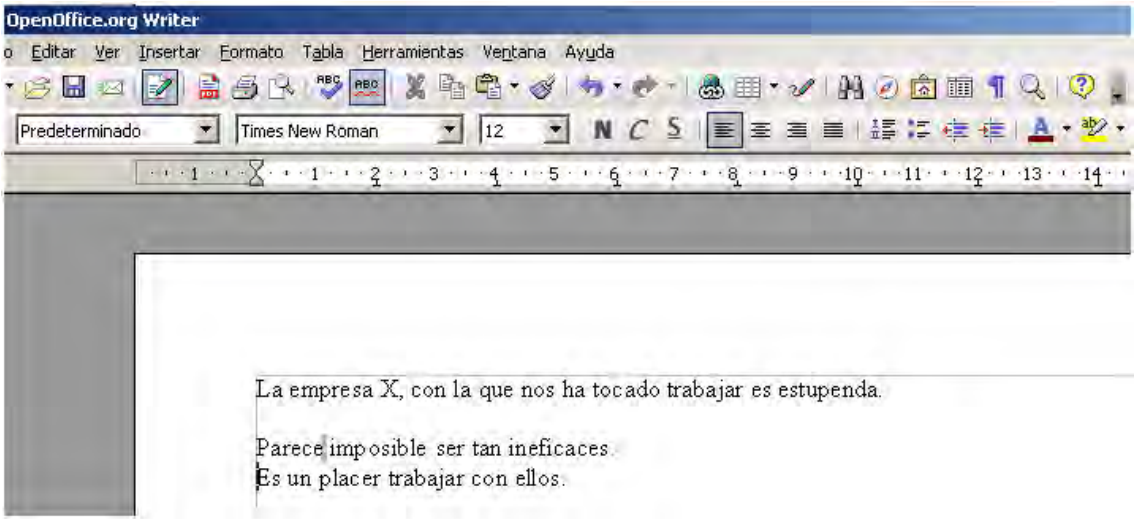


*Figure 16: Document displaying hidden paragraphs*

### 2.9 HIDDEN INFORMATION DUE TO THE FORMAT

Another type of hidden text or content is the one that is not visible due to the document format: for example, other content, such as an image, overlaps with it or the text is in the same color that the document background. Of course, this kind of content should be carefully reviewed before publishing the document.

### 2.10 NOTES, HEADERS, FOOTNOTES AND COMMENTS

In an OpenOffice document there are a number of places where you can enter information that may go unnoticed in subsequent revisions. For example, in headers and footnotes, online annotations or comments, that can be entered using the "Notes" option in the "Insert" menu. These notes, unless you specify it, are not included when the document is printed or exported, for example, to PDF format, so it is easy to forget this information in the reviews. It has to be considered that some elements can be defined as "not printable"; therefore, a detailed revision of a document should not be limited to reading a printed version.

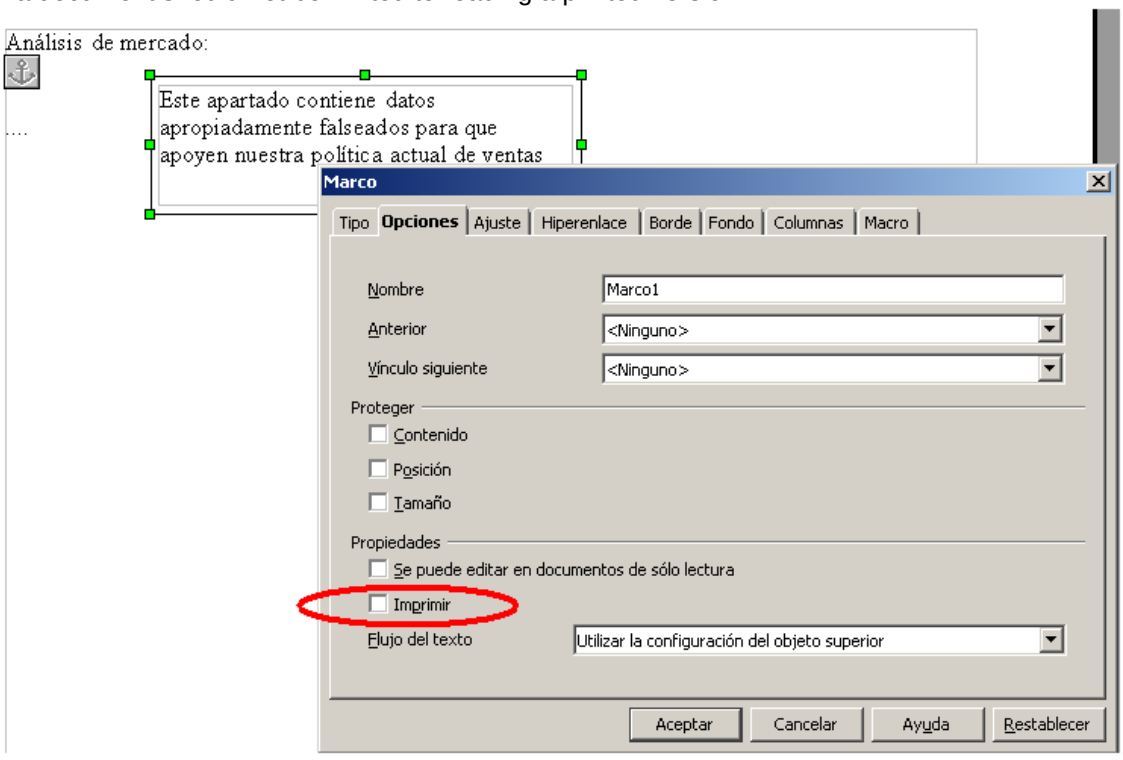


*Figure 17: Defining a document as printable or not printable (Spanish version, what a great language!)*

### 2.11 CUSTOMIZED METADATA

Metadata, as it was discussed in the introduction of this document, are not harmful themselves, and on the contrary, they can be very useful for certain applications. In OpenOffice, the user is able to include customized Metadata in his/her documents and to add information to the document using the "Properties" option from the File menu. In addition to the customized Metadata, the document may store information if it is created from another previous document, inherited from this one.

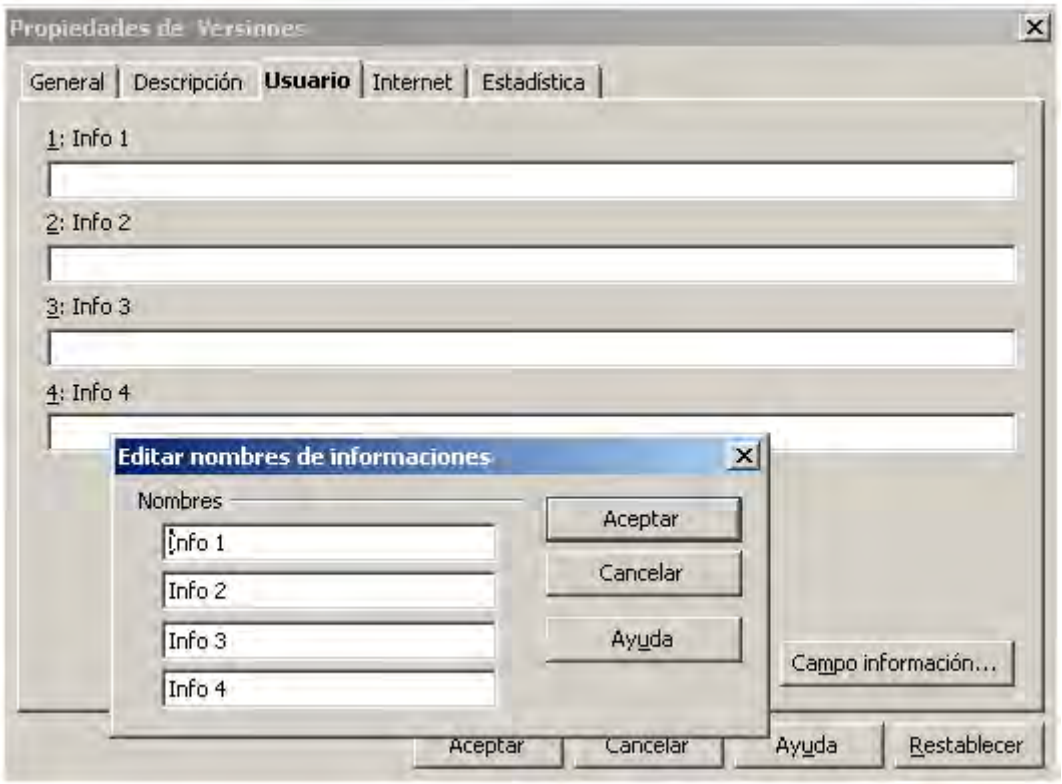


*Figure 18: Customized metadata*

Sometimes, this customized information is used as a working tool in the process of document elaboration and may include corporate or personal opinions, more or less politically correct, identifications and other personal data or references to documentary sources. Of course, all this information must be reviewed before the document publication.

### 2.12 DATABASES

The combination of documents with databases must be considered too. One of the most important functionalities provided by Office applications nowadays is the ability to generate models that, combined with databases, allow customized and automatic documents generation. These models, designed from mail merging, deserve a special consideration, since they contain information that allows describing the database they are taking the information from. All the information related to the database can be found in the settings.xml file. There is information about the name of the database and about the table used for the combination.

```
<config:config-item config:name="CurrentDatabaseDataSource"
config:type="string">Referencias</config:config-item>
<config:config-item config:name="CurrentDatabaseCommandType" config:type="int">0</config:config-
item>
<config:config-item config:name="CurrentDatabaseCommand"
config:type="string">Contactos</config:config-item>
<config:config-item config:name="PrintDrawings" config:type="boolean">true</config:config-item>
```

*Figure 19: Information related to database in settings.xml*

And in content.xml file we can find the name of the database, the table and the fields:

```
…
<text:p text:style-name="Standard">
<text:database-display text:table-name="Contactos" text:table-type="table" text:column-
name="nombre" text:database-name="Referencias"><nombre></text:database-display>
</text:p>
<text:p text:style-name="Standard">
<text:database-display text:table-name="Contactos" text:table-type="table" text:column-
name="direccion" text:database-name="Referencias"><direccion></text:database-display>
</text:p>
<text:p text:style-name="Standard">
<text:database-display text:table-name="Contactos" text:table-type="table" text:column-
name="clave" text:database-name="Referencias"><clave></text:database-display>
…
```

*Figure 20: Information related to database in content.xml*

However, information regarding the connection to the database is not in the ODF document. This information could show the path to a database file or the credentials to access a server, and is stored in a user's profile file called DataAccess.xcu, that should be especially protected by the user.

*C:\Documents and Settings\USER_ACCOUNT\Program data*
*\OpenOffice.org2\user\registry\data\org\openoffice\Office\DataAccess.xcu*

Although connection's credentials to database are not published, the information stored with the document may be enough to help a potential attacker to prepare his attacks to the database, directly or through SQL Injection techniques on the company's website.

### 2.13 VERSIONS OF DOCUMENTS

Like other Office packages, OpenOffice allows saving different versions of the same document. This feature is extremely useful in collaborative work environments, allowing the evaluation of the document changes and, if necessary, restoring the previous state after a mishandling. In the File menu, the "Versions" option is to save the current version of the document, and to save a new version of the document every time.

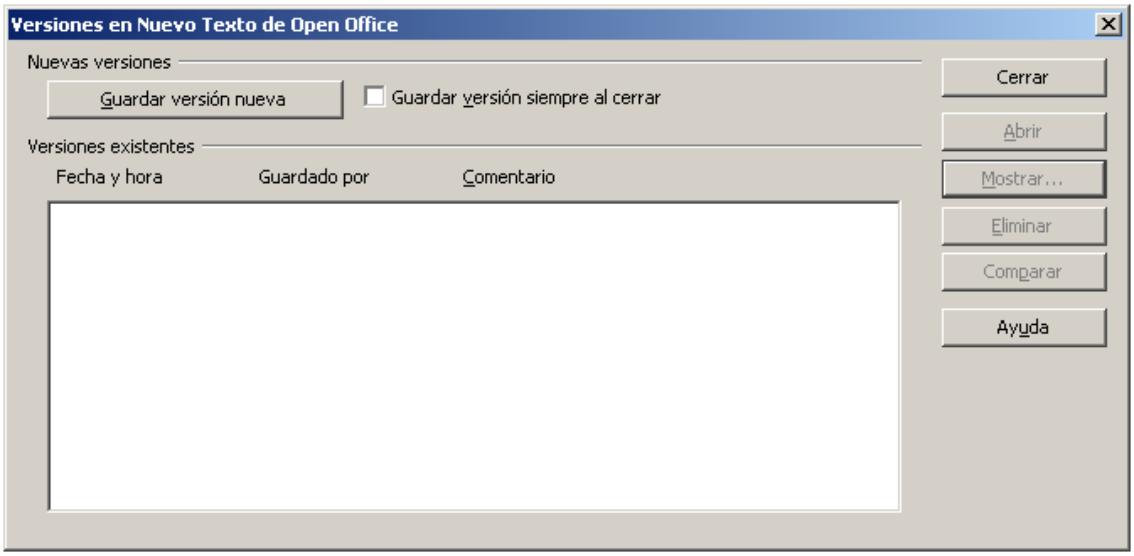


*Figure 21: Document's versions (Software in Spanish)*

Within an ODF document that contains different versions we can find important information. First of all, a file called VersionList.xml with information about who saved each different version and when:

```
<?xml version="1.0" encoding="UTF-8" ?>
<VL:version-list xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:VL="http://openoffice.org/2001/versions-list">
<VL:version-entry VL:title="Version1" VL:comment="Versión guardada automáticamente"
VL:creator="MiNombre MiApellido" dc:date-time="2008-08-13T00:39:22" />
<VL:version-entry VL:title="Version2" VL:comment="Versión guardada automáticamente"
VL:creator="MiNombre MiApellido" dc:date-time="2008-08-13T00:41:53" />
…
</VL:version-list>
```

*Figure 22: VersionList.xml*

Secondly, all the different versions of the document are stored in a folder called "Versions". For each of them, we can find the complete structure of a document in OpenOffice, ODF, that is, each version contains the files meta.xml, settings.xml, content.xml, etc...

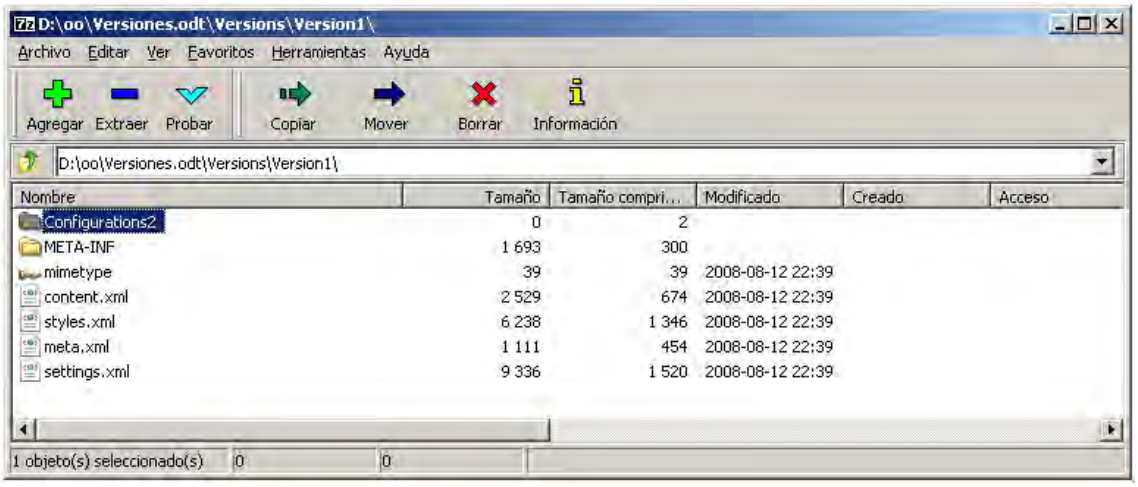


*Figure 23: Versions folder*

### 2.13 ANALYSIS AND CLEANING TOOLS

There are some tools to analyze and remove Metadata in OpenOffice, such as 3BOpenDoc or 3BClean, but these tools only remove the file meta.xml, leaving the information about printers, possible internal servers and data connections to databases in the files settings.xml and content.xml.

It is true that the meta.xml file is especially important because, as the documentation says, this file is not going to be encrypted even when you save the file as password protected, but in terms of security the rest of the Metadata is equal.

Therefore it is very important to have a tool capable of analyzing all the information stored in these files and providing a convenient and user-friendly environment for all users in an organization to clean al the Metadata files in OpenOffice.

OpenOffice itself offers the option to "Remove personal information when you leave," but this option does not remove the information from the operating system, printers, the version of the product, the path to the template. And we have seen that this information can show hidden routes or addresses from internal servers and of course, connection information to databases

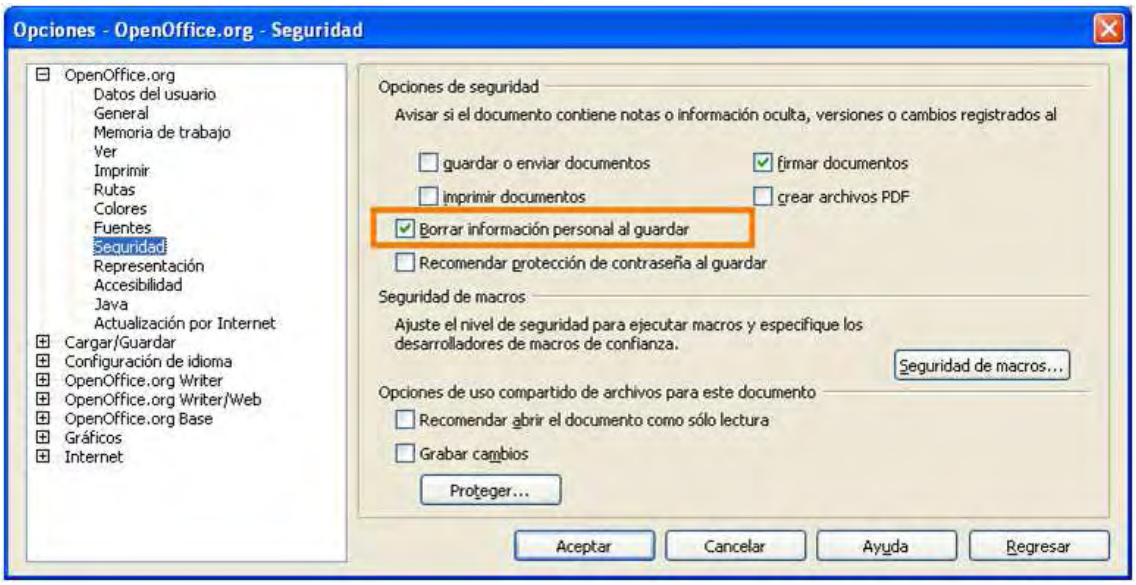


*Figure 24: Delete all personal information when save the document (Software in Spanish)*

As a conclusion, the "Delete all personal information" option and the use of the available tools eliminate some metadata but not all the hidden information within an OpenOffice document.

A last alternative is to use tools to recover damaged documents. Using the Recovery for Writer, for example, the information stored in the meta.xml and settings.xml files, disappears. However, the information about the structure of the database and the connection in the content.xml file does not disappear, and in documents created with templates, the look of the document may change. Therefore, it is not a definitive solution.

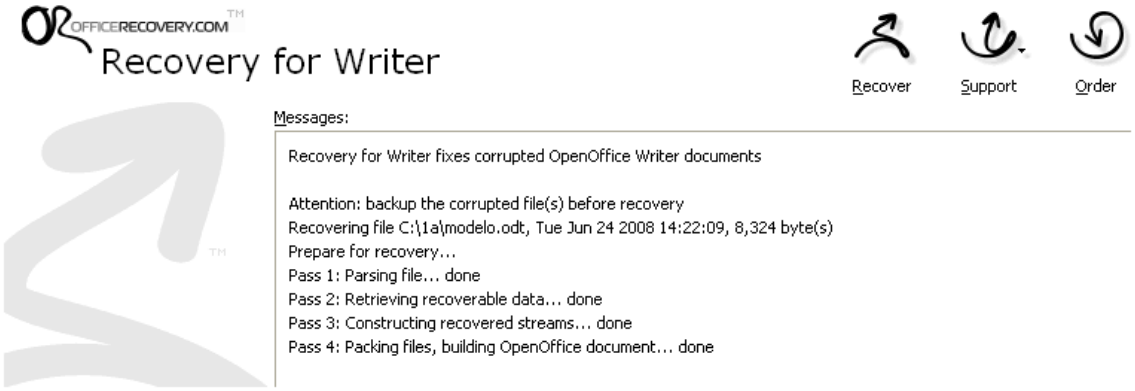


*Figure 25: Recovery for Writter*

### 2.14 OOMETAEXTRACTOR

As a better solution we conclude to develop a tool to extract and clean metadata and hidden info in OpenOffice documents. This tool is available under a Microsoft Public License in Codeplex web site. It has been developed in .Net so it needs .NET Framework and it has been tested only in Microsoft Windows operating systems.

This tool allows users to analyze not only an ODF document but also a complete folder full of ODF documents in ODT, ODS or ODP formats. All metadata and hidden info are shown and can be exported in a text file. Moreover, this tool cleans all documents, even templates, links to documents, printer configuration and customized metadata.

In order to create a company's policy with metadata this tool has an option to set up a template for metadata. This means, what to do whit some kind of metadata. For instance, it would be desirable set up company's name in company metadata, or a fixed author for all documents.

This can be done easily with OOMetaExtractor. This tool is available for download along with its source code at http://www.codeplex.com/oometaextractor

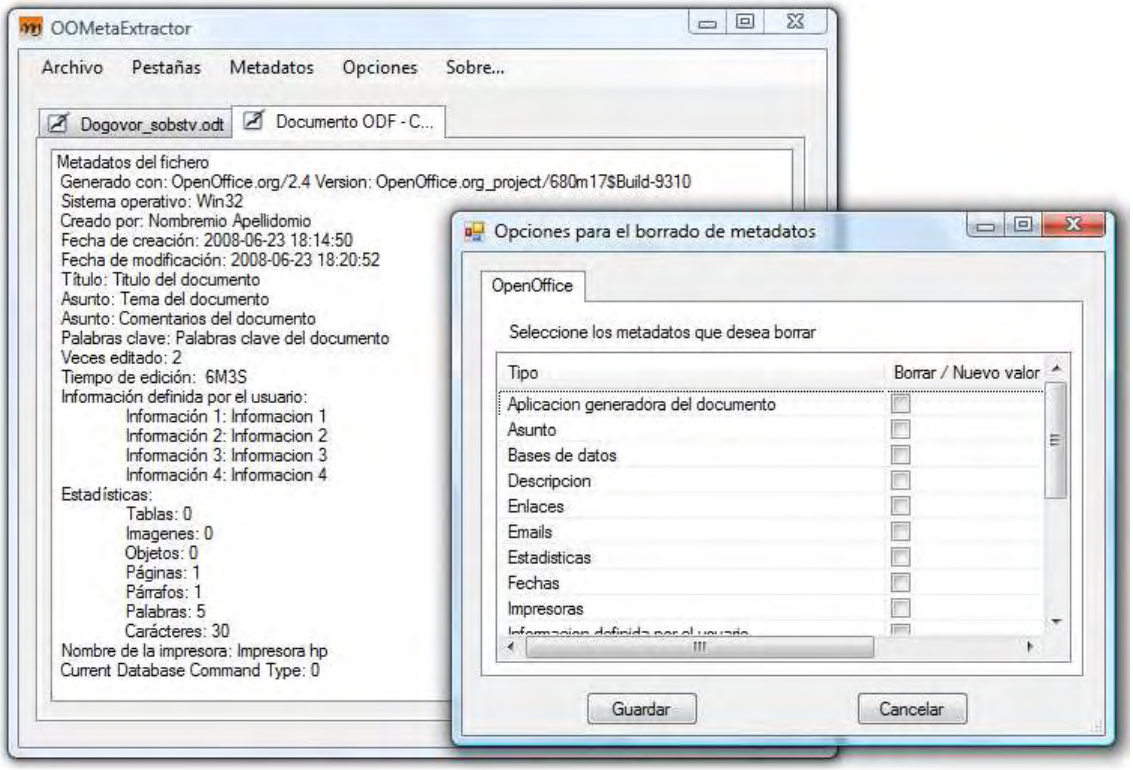


*Figure 26: OOMetaExtractor (yes, it´s in Spanish)*

## 3 METADATA AND HIDDEN INFORMATION INSIDE MICROSOFT OFFICE DOCUMENTS

During the Microsoft Office installation, a dialog lets the user input information about him or her. From that moment on, the information the user provided will be added to each and every document created, or edited, by that user with this software.

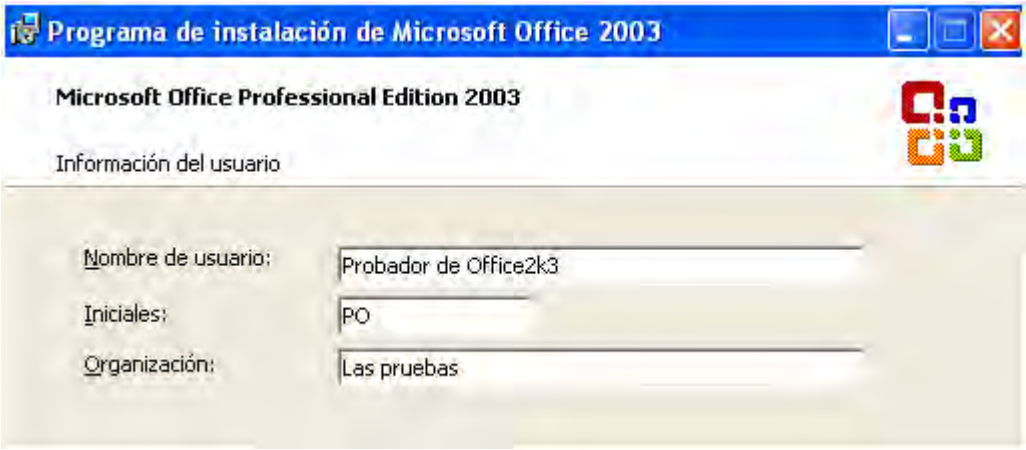


*Figure 27: User Information in Microsoft Office 2003*

In multi-user environments, the same product is often used by different users on the same computer. When a user runs an Office application for the first time, a new dialog appears asking for information about him or her. And, again, this information will be added to every document the user edits o creates.

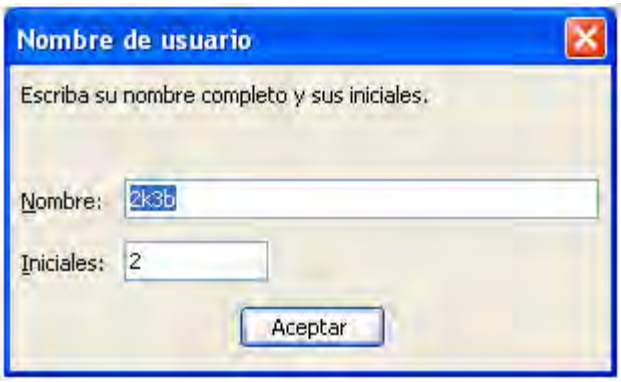


*Figure 27: Another User stars Microsoft Office 2003*

And this information can pose a risk, above all because the default value for the "Name" field is the user account name.

### 3.1 DOCUMENT PROPERTIES

Upon document creation, authors can explicitly assign metadata to it, that is, creator can introduce a short description, keywords, and their department o whatever that may be suitable or useful. This data gets stored indefinitely inside the document file. When a document containing metadata is used as a template in order to create new documents, these ones will inherit this information.

Document metadata is customizable, so they may contain any attribute and value the author could add. This must be kept into account when publishing documents created in corporative environments, because an inappropriate metadata can damage the organization image.

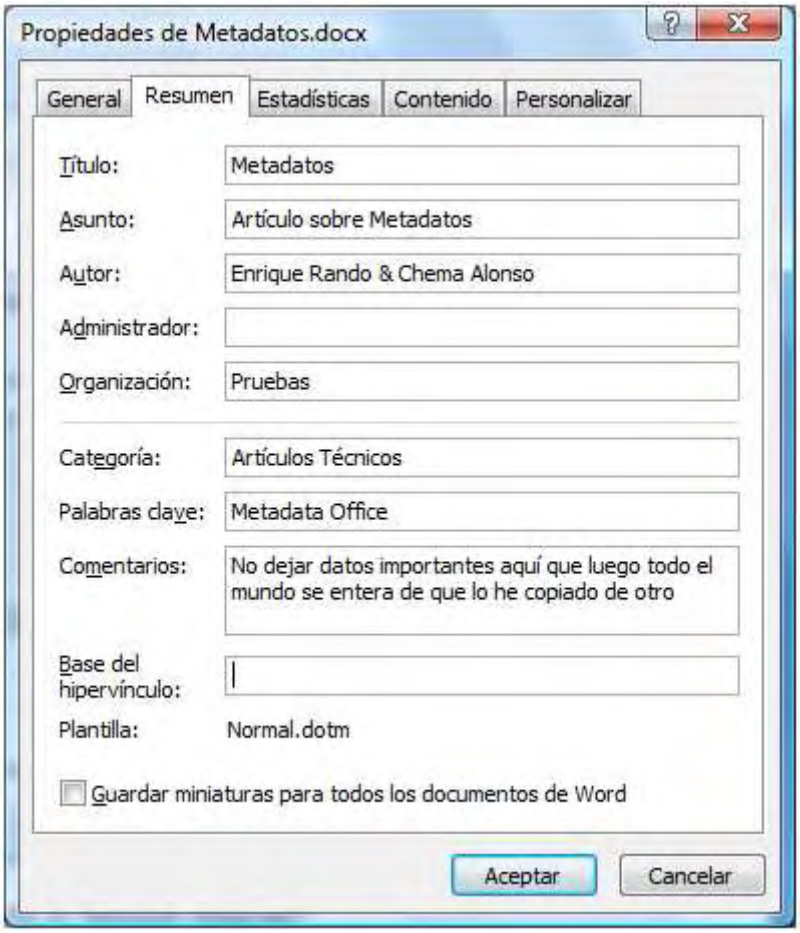


*Figure 28: Document´s properties (all in Spanish….)*

### 3.2 EMBEDDED FILES

Microsoft Office allows users to embed images and other documents into the documents they create. These embedded documents may contain their own metadata, thus being a potential source for information leaks.

The next example shown an image file created with GIMP, a graphic document creation program. That image contains EXIF information, readable with any EXIF extraction tool. It may be seen that the image has a metadata attribute showing the program used. It also has a thumbnail for the image inside the same file.

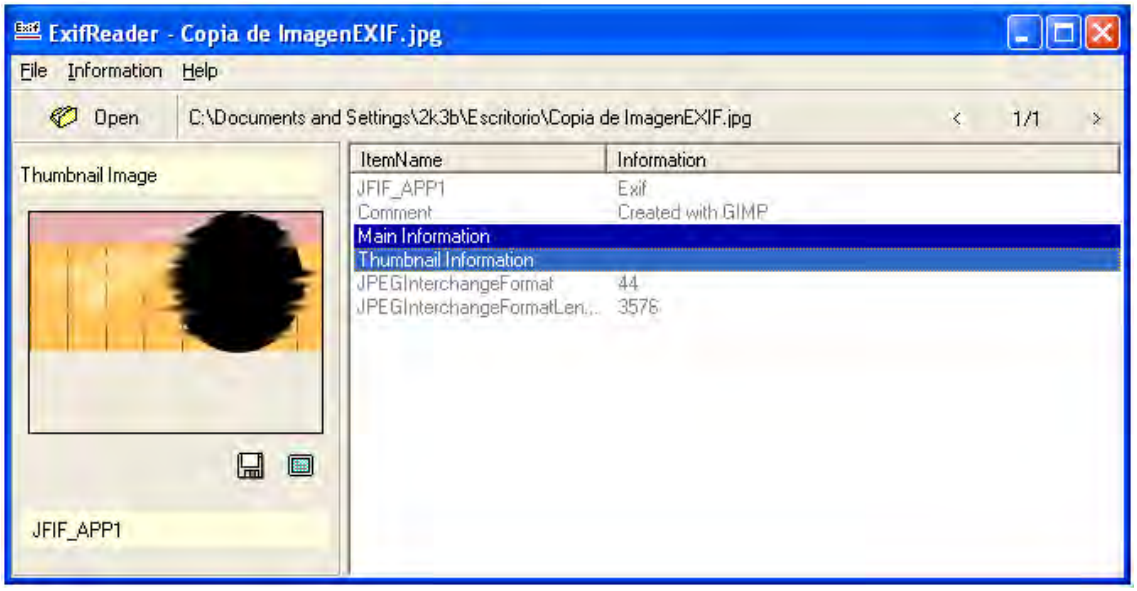


*Figure 29: EXIF Metadata in an embedded image*

Let's embed this image inside a Microsoft Word 97 document, using the "Insert Image" option from the "File" menu. Then, using a hexadecimal editor, the image metadata can be read.

### 3.3 EXTRACTING EMBEDDED FILES

It is easy to extract these embedded files from Microsoft Word DOC, Excel XLS or PowerPoint PPT files. Just by saving the document in HTML (web page) format, these programs extract embedded files and store them as independent files.

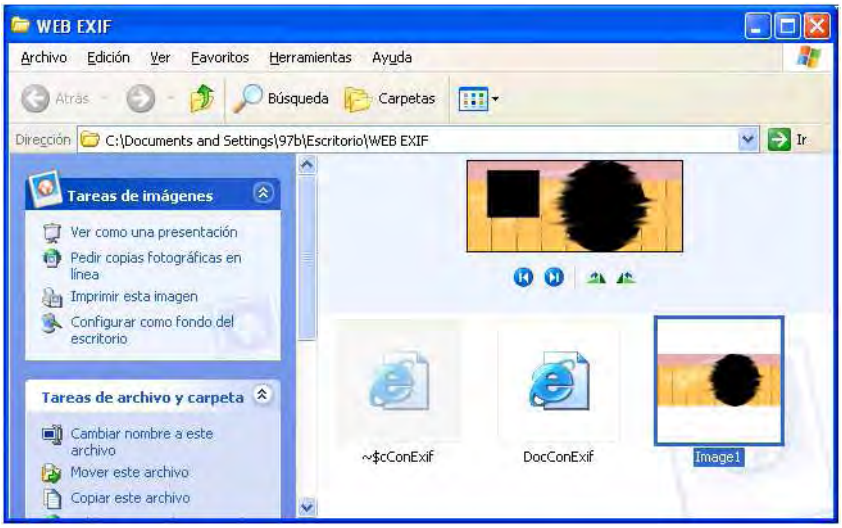


*Figure 30: Extracting embedded files*

It can be seen that EXIF information has not been modified. In this example, the thumbnail looks different from the image, showing that some editing work has been made on it.

This metadata preserving behavior can be found in Microsoft Office versions. They don't modify EXIF information unless the author uses the "Modify Image" option and then, saves the document.

Microsoft Office 2007 introduces a new file format, called OOXML which will become to ISO DIS 29500 in next Microsoft Office versions. DOCX, XLSX, and PPTX files are ZIP compressed archives in which embedded files are stored as independent items. These items can also be easily extracted.


*Figure 31: Files inside an OOXML file*

### 3.4 REVISIONS Y MODIFICATIONS

Users involved in document sharing and workflows find Microsoft Office "Revision" feature especially useful. This feature allows more than one user to work on the same document, while keeping track of who made each change, so that previous document states can be recovered.

But when the document is made available to the public, this data is no longer useful and can turn into compromising information.


*Figure 32: Changes can be seen (Image in Spanish)*

As the image shows, older contents are shown in red color. Final document view gives a positive result, while the older one gave a negative one. This could be seen by any user, only by selecting the "see original document".

### 3.5 NOTES, HEADERS AND PAGE FOOTINGS

There are other places where non intended information may appear, such as notes and page headers and footers, or presentation annotations. They can contain reference codes used inside the organization, names of users that worked on the document or file paths. All this information must be taken into account before the document is published.

### 3.6 ELEMENTS HIDDEN BY THEIR FORMAT

In a Microsoft Office document, an image can be hidden by other ones above it. Template elements may contain undesired data that afterwards can be hidden by document text or images. Some text can be of the same color than the background. All these items are not visible, but they remain within the document and therefore can be use to extract information.

Errors and negligent or malicious behaviors may cause this kind of data be part of the documents the organization publishes or send out.

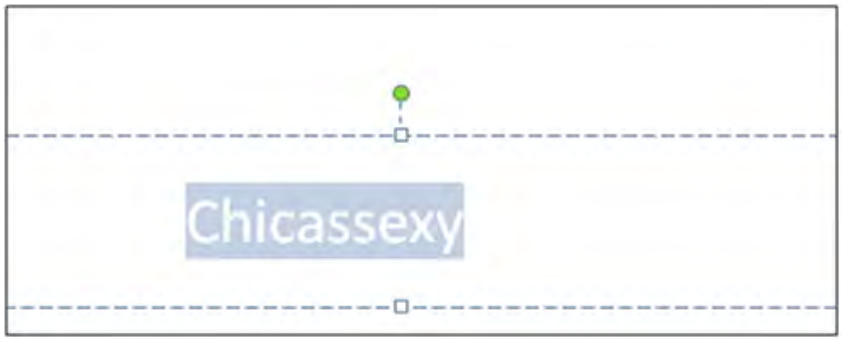


*Figure 33: Data hidden by document's format*

### 3.7 OTHER PLACES WHERE TO LOOK FOR DATA

The list of places where undesired data may appear is very long: comments, style sheets, hyperlinks to Intranet servers, or even parameters or variable names in VBScript macros… All this data can be easily retrieved using a hexadecimal editor or a string extraction program.

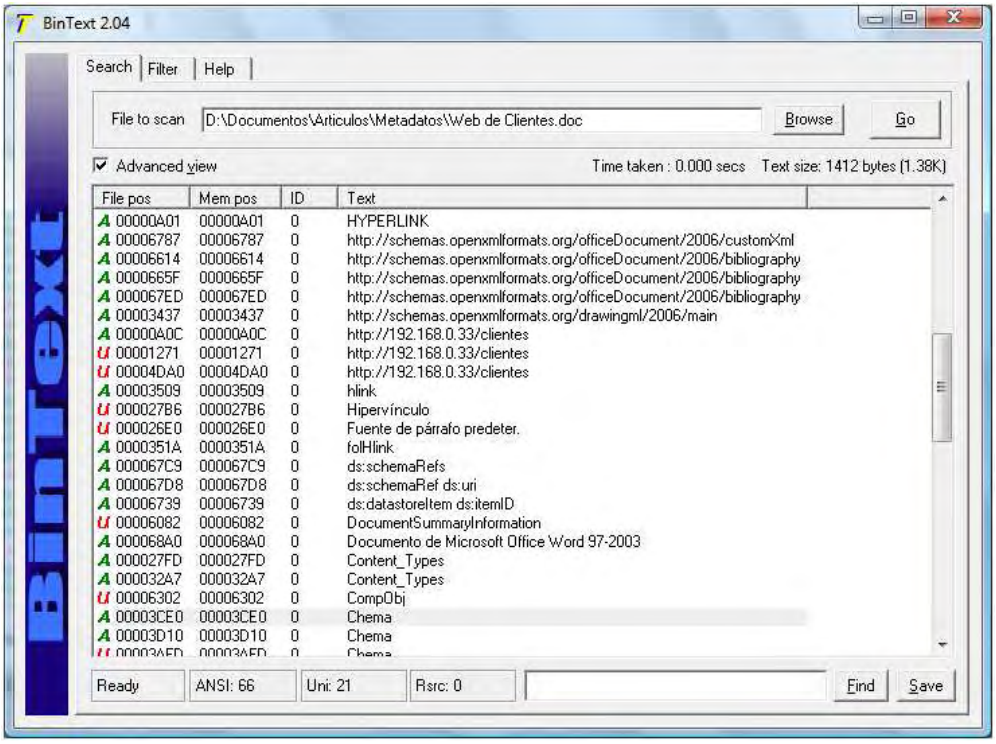


*Figure 34: Strings extracted from an OOXML file using bintext*

### 3.8 HIDDEN METADATA

So far, this part about Microsoft Office has dealt with metadata and information easily accessible, and sometimes editable, by the user. But there is another kind of data that is stored inside documents and is not available to the user.

These metadata are used by Microsoft Office in order to perform its own tasks. And they may contain compromising information such as software versions, authors, revision history, the last person who edited it and when he did it, the last time the document was printed, which printer was used, total editing time for the document, information about e-mail messages including e-mail addresses, and even, in some earlier versions of Office, a Global Unique ID that identifies the computer on which the document was edited.

```
mimetype - application/msword
revision history - Revision #1: Author 'EL USUARIO' worked on '\\servidor\compartida\Doc01.doc'
revision history - Revision #0: Author 'EL USUARIO' worked on 'C:\Documents and
Settings\usuario97\Escritorio\Doc01.doc'
company - LA ORGANIZACION
paragraph count - 1
line count - 1
last printed - 2008-05-19T09:36:0
last saved by - EL USUARIO
character count - 225
template - Normal
creation date - 2008-05-19T09:21:0
title - Prueba de documento de Word97
word count - 39
page count - 1
creator - EL USUARIO
date - 2008-05-19T09:46:0
generator - Microsoft Word 8.0
```

*Figure 35: Extracted document's metadata using Libextractor*

### 3.9 DATABASE CONNECTIONS

Information stored in a particular file depends on the Microsoft Office version used and the file format. Sometimes, if document is using an external source to be constructed even information about databases and ODBC drivers can be retrieved. The following image shows a SELECT query, configured ODBC drivers, the database server, database itself and the password too.

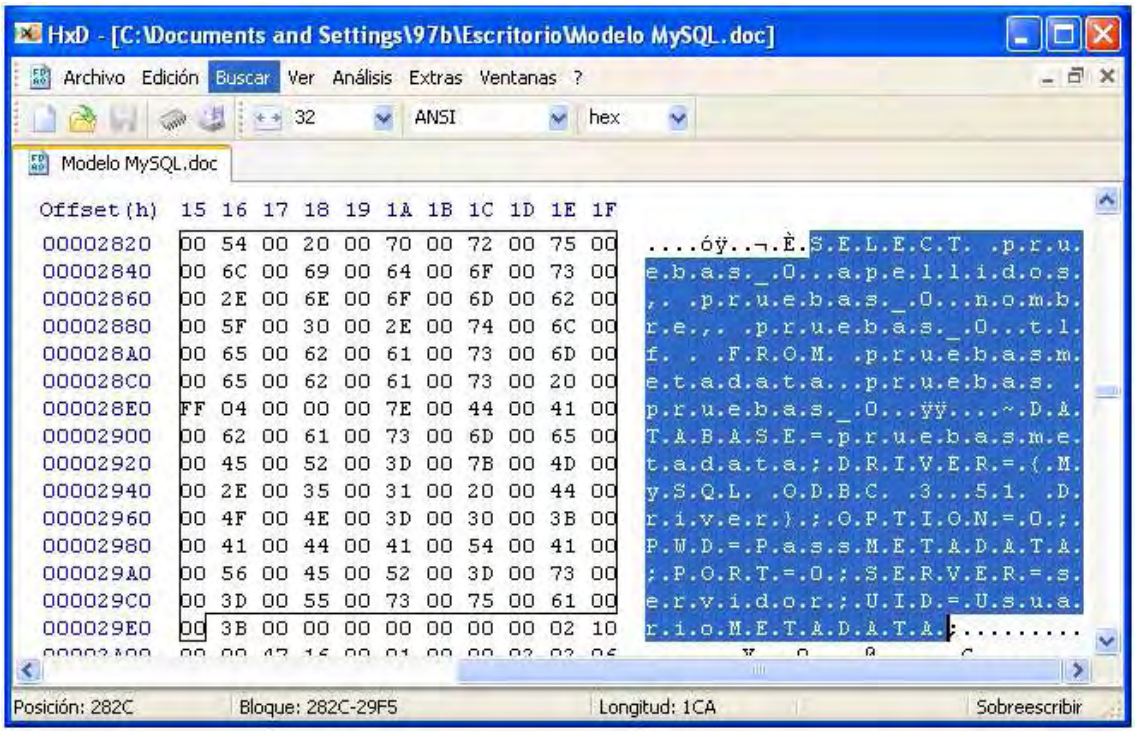


*Figure 36: Database info*

Putting it in clear text in figure 36, we have:

*SELECT pruebas_0.apellidos, pruebas_0.nombre, pruebas_0.tlf FROM pruebasmetadata.pruebas pruebas_0*
*DATABASE=pruebasmetadata*
*DRIVER={MySQL ODBC 3.51 Driver} OPTION=0*
*PWD=PassMETADATA*
*PORT=0 SERVER=servidor*
*UID=UsuarioMETADATA*

A hexadecimal editor is, again, all it takes to read this data from a document. As can be seen, this is a special document created to generate documents changing some special parts which are coming from a database repository.

### 3.10 PRINTERS

As it was seen before, revision history provides information about user accounts and paths that may relate to server names and shared resources. Another compromising hidden piece of information that can be found inside documents is printer data.

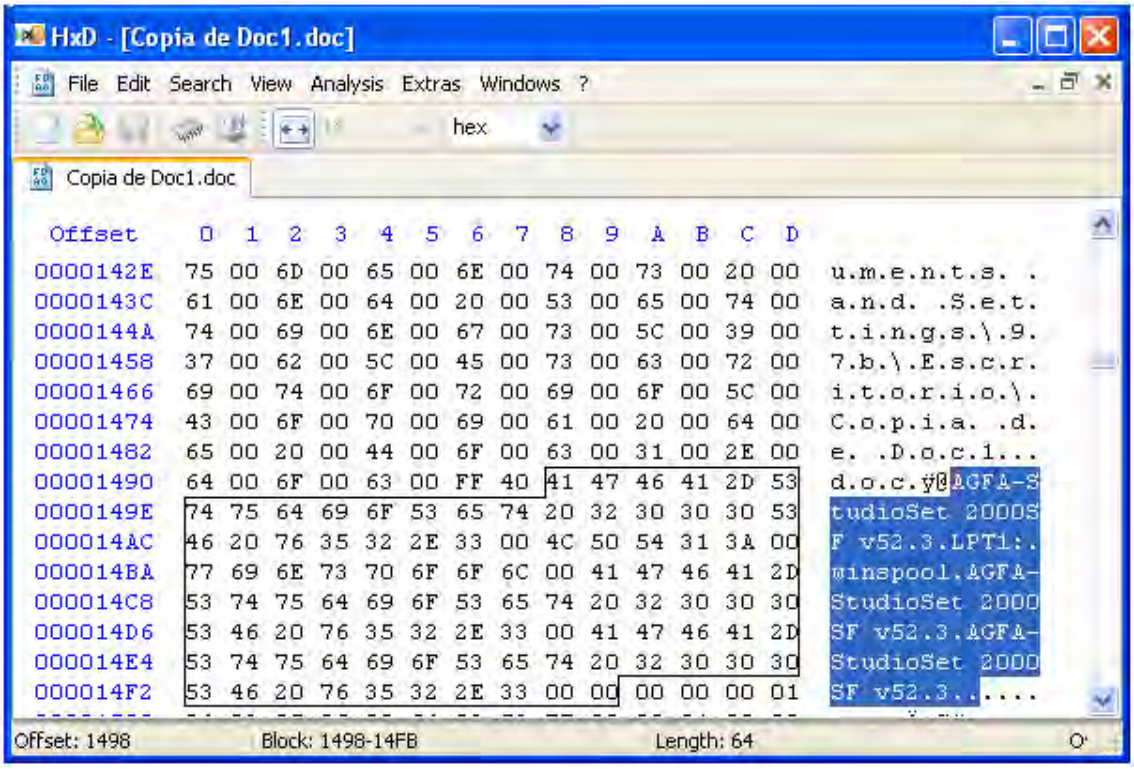


*Figure 37: Information about printer*

As in ODF documents, sensitivity of this information depends on the way printer had been configured. When the printer is shared on a server, it can appear in UNC format, thus providing a server name and a shared resource and even the server's IP address, revealing the way the internal network has been addressed.

Although this is quite important by itself it is giving more information to a potential attacker just because if document´s creator is using that printer then it means that user has access to that resource so, of course, that user is a valid one in server's. Hence, at the end, it´s also possible to discover information about network's ACLs.

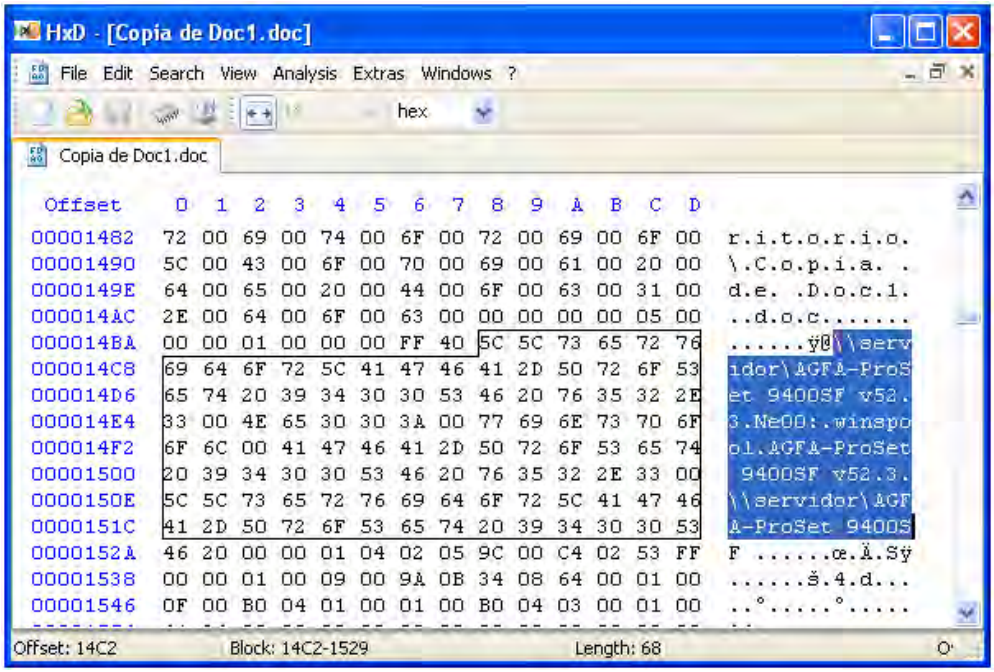


*Figure 38: Printer in UNC format*

Microsoft Office versions prior to Office 97 seldom include this information into document metadata. But all versions do update it if they find it, changing its value to the name of the printer currently in use, even if the document is not printed: a simple edition and saving is all it takes. Hence, documents which have been used by the organization since those old Office 97 times must be carefully inspected for printer data.

### 3.11 DOCUMENT ANALYSIS AND CLEANING

Cleaning Microsoft Office documents before sending or publishing them is a must, since they may contain huge amounts of undesired data. The ultimate tool for accomplishing this task is the “Inspect Document” option available in Office 2007. It searches for all metadata and hidden information, including printer data or revision history, and gives the user the option to remove it all from the document. No matter what version of Microsoft Office was used no create and edit the document, “Inspect Document” will clean it with only a few clicks.

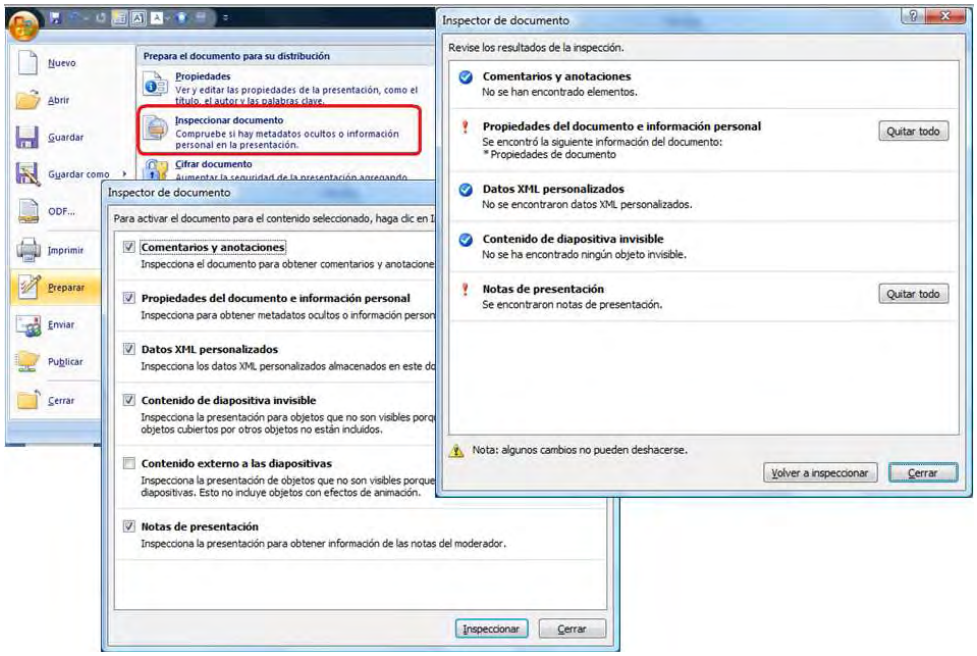


*Figure 39: Preparing document in Microsoft Office 2007 (yes, it´s Spanish)*

Previous versions of Microsoft Office didn't provide this option. For that reason, Microsoft released a plug-in to improve software with cleaning options. This tool is known as RHDTool and it´s available for downloading in the following URL:

http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=144e54ed-d43e-42ca-bc7b-5446d34e5360


*Figure 40: RHDTool*

Also, Microsoft has published some guides to help users to minimize the amount of metadata in their documents (their URLs are in the References section of this paper).

Another option is using third party tools, like Metadata Extractor o Doc Scrubble. It must be said, anyway, that from the tests made it can be deduced that sometimes these tools don't clean all metadata and hidden information. For example, the following image shows DocScrubble options, and, for instance, although a document was cleaned using DocScrubble, it keeps containing printer data.

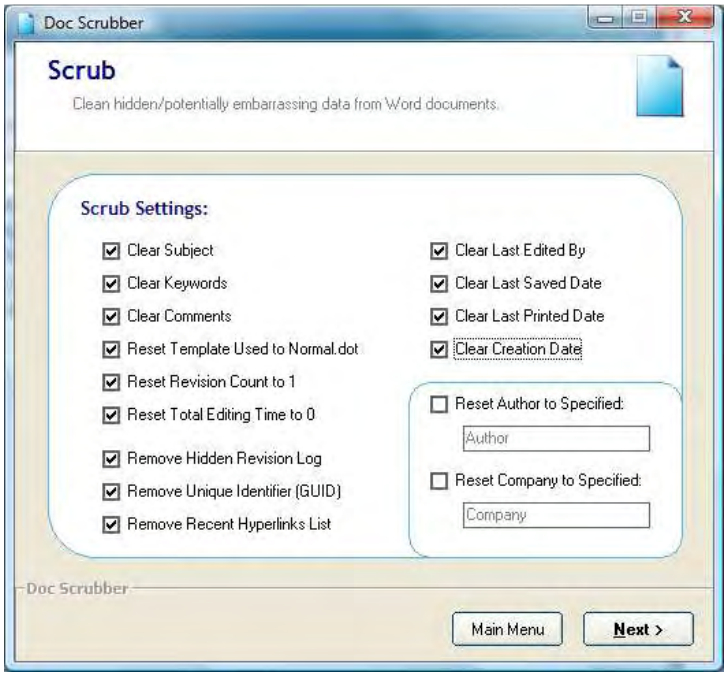


*Figure 41: Printer info remains inside the file*

It has been said that converting Microsoft Office to other formats may be a solution to the problems stated in this article, in particular, converting them to PDF format. But PDF files store information too, both in the "Data Dictionary" and in XMP streams.

These metadata may include user accounts, file paths, URLs pointing to Intranet servers, e-mail message headers including e-mail addresses, information about the Operating System and more.

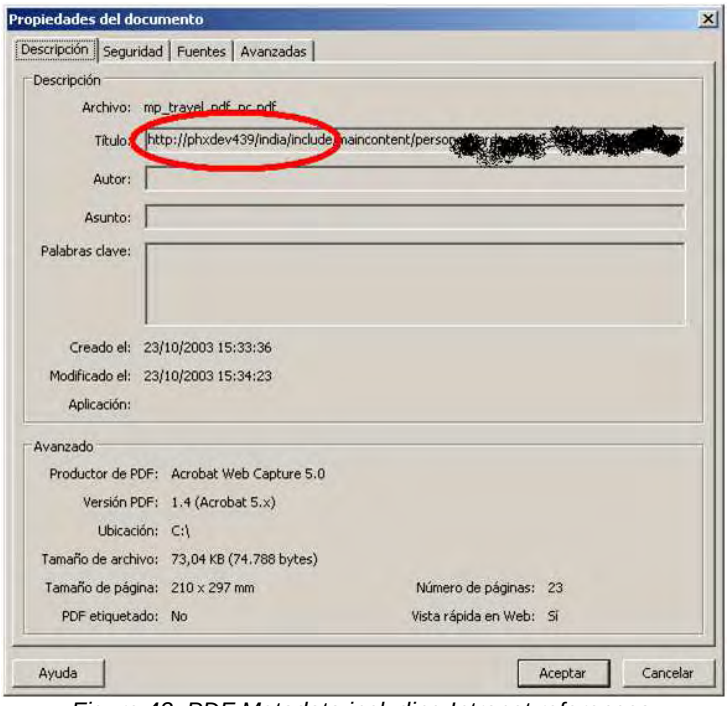


*Figure 42: PDF Metadata including Intranet references*

But even converting documents to TXT format may not be enough, even if they contain no metadata at all. When a document is published in the corporate web, it becomes available to web search engines, as Google, MSN Search or Yahoo. And this search engines create new metadata about them: they give them a title, provide an abstract of their content as part of the results and store a cached version.

As that search engine generated metadata is created based upon the document content, any lost data that the document contains may turn into metadata.

## 4 METADATA EXTRACTION TOOLS

There are several tools to retrieve metadata from documents, depending on their format. Maybe, *libextractor* is the best known of them all. It features a library of functions to access document's properties and a standalone program, called "*extract*" which parses and extracts metadata from several file formats, including Microsoft Office documents, ODT, PDF and many more. The support for Microsoft Word DOC files includes extraction of the revision history, being one of the few tools freely available to accomplish this task.

*Metagoofil* is a program that downloads documents from web sites and afterwards extracts metadata from them using *extract* tool, making the process of fingerprinting corporate networks easier.

*Exiftool* is another open source metadata extraction tool. Originally designed to extract EXIF information from image files, it can process a huge amount of file formats, including again Microsoft Office Documents. When present, it can retrieve GUID information from old versions Microsoft Word files. Another remarkable feature of Exiftool is the way it deals with PDF documents, being able to extract both Data Dictionary and XMP metadata. It has an option that allows in deep scanning for XMP metadata even if it has been disassociated from the PDF file main tree structure. Both *extract* and *exiftool* are command line utilities.

There are tools to extract metadata through a GUI, like *ExifReader*. Anyway, this kind of programs tends to be less flexible and powerful than their command line alternatives.

But all these programs deal mostly with metadata and don't pay much attention to hidden information, with the exception of *extract* and its revision history processing. Neither printer information, templates paths, nor database information can be obtained by them. To solve this problem, new tools have to be made. And that's why the FOCA tool was created.

FOCA uses web search engines to locate documents, download them and extracts metadata and hidden information from them. It can retrieve information from several file formats, including Microsoft Office documents, ODF, Word Perfect files and more, and the range of information obtained is wider that the one achieved with any other tool mentioned here.

## 5 SEARCH ENGINES

When a document is published in the corporate web, it becomes available to web search engines, such as Google, Live Search or Yahoo. These search engines create new metadata about them: they give them a title, provide an abstract of their content as part of the results and store a cached version.

As that search-engine-generated metadata is created based upon document's content, any lost data document contains may turn into metadata. That is even worse if document does not contain title or customized metadata helping search engines to create a short description of it.

With a little Google Hacking work, addressing metadata fields as the "title" through using searching options as "intitle", compromising information can be retrieved. There is no need for even downloading documents hence organizations may not even notice someone is accessing sensitive data.


*Figure 43: Fbi.gov users gathered by Google*

Of course, this kind of data can be found in documents created or edited on Linux, and all Unix flavors, systems too:


*Figure 44: *NIX users*

All this information becomes available to anyone who has Internet access. And it may remain available even after documents were modified or removed from web site due to the cache feature most web search engines provide.

Cached copies from documents may appear tagged with the words “Cache” or “HTML version”, and they are stored by the search engine systems themselves, thus giving little control to website owners and administrators on their contents and access rights. Of course, search engines companies provide tools which allow webmasters to remove contents from their indexes, but those tools may be useless for massive document management. Besides, monitoring accesses through search engine caches is, at least, a taught task for organizations and website administrators.

## 6. FOCA

Just as a proof of concept, FOCA is been developed. FOCA, which stands for “Fingerprinting Organization with Collected Archives” is an automated tool for downloading documents published in websites, extracting metadata and analyzing data. Now it´s an on growth project which is being improved day by day.

FOCA searches for links to documents using Google and Microsoft Live Search engines. This tool is not using any special Google key to access the API, so if Google or Microsoft Live asks for a CAPTHA value then FOCA will stop and show up the CAPTCAH, waiting for it to continue.

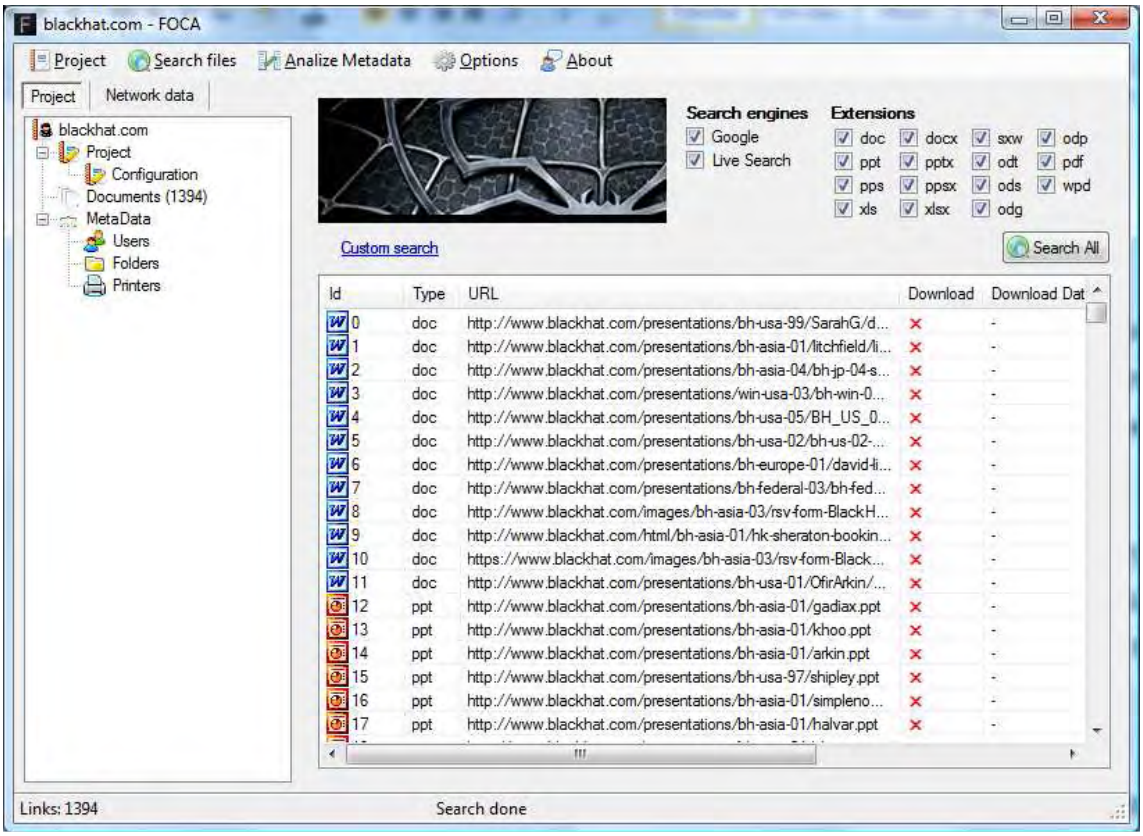


*Figure 45: 1394 office documents published in Blackhat web site*

A big website can store between three and five thousand Office documents so, in a normal search for links maybe between zero and five CAPTCHAs can be required (depending on the Google and Microsoft security policies in that moment). After collecting all the links, all documents can be downloaded using a multi thread engine to retrieve all of them as fastest as possible.

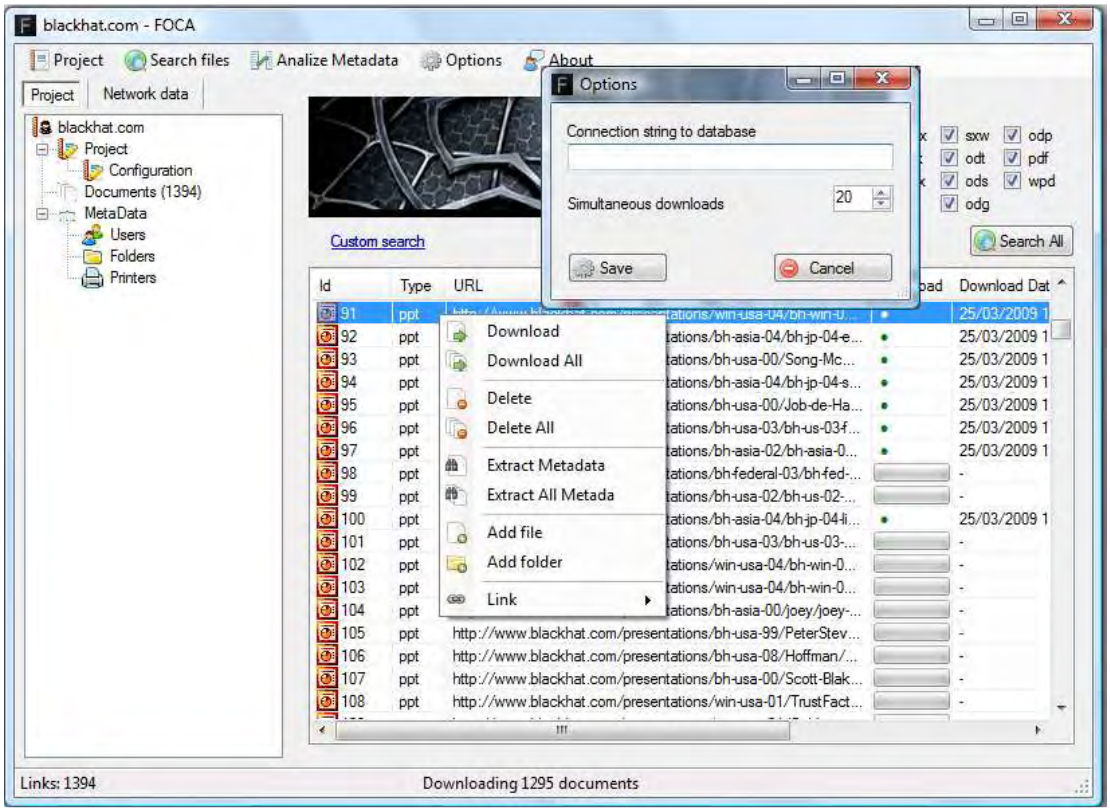


*Figure 46: FOCA downloading files*

Once retrieved, FOCA allows extracting metadata from all documents. Today FOCA supports DOC, XLS, PPT, PPS, DOCX, PPTX, PPSX, XLSX, SWX, ODT, ODS, ODP, PDF, and WPD documents. In the following image the metadata extracted from a public document is shown. In this case, the document is a (great) David Litchfiled work.

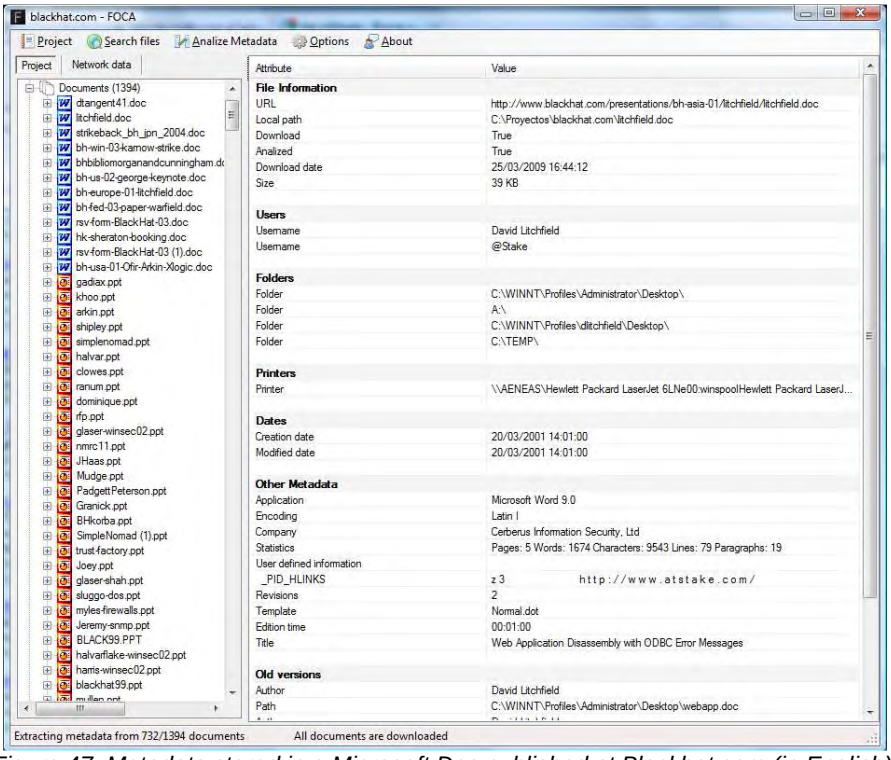


*Figure 47: Metadata stored in a Microsoft Doc published at Blackhat.com (in English)*

Once all documents are analyzed FOCA will collect three special lists. First one with discovered users, second one with paths to files and the last one with printers. Besides, using FOCA is easy to track where the metadata is stored in order to analyze deeply a real environment.

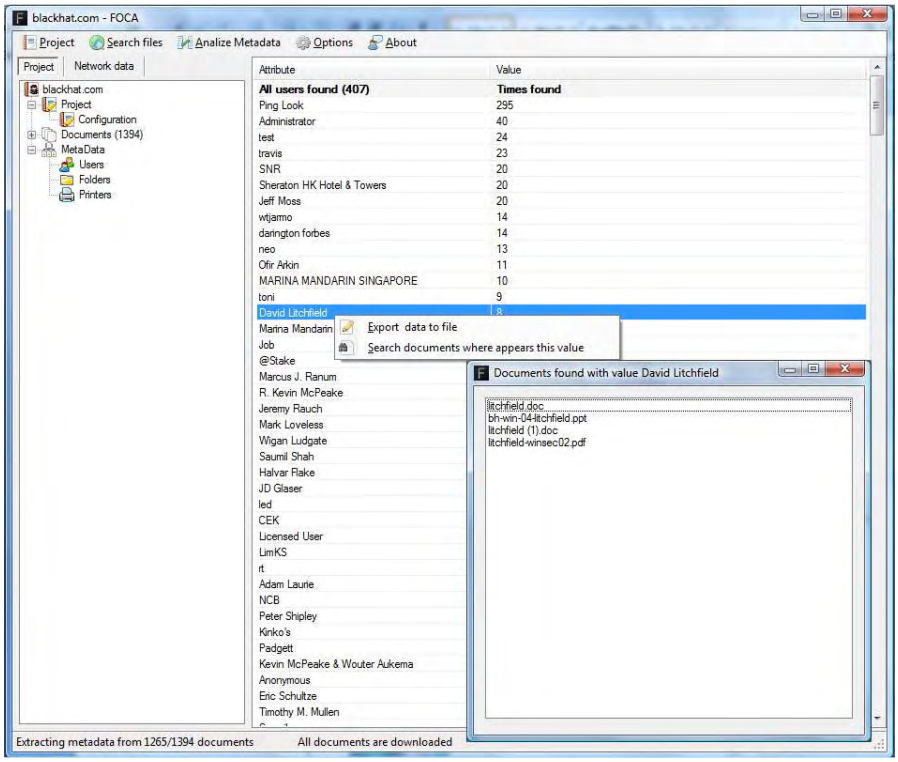


*Figure 48: Tracking Metadata in files*

FOCA will also search for new servers in the domain name server using Google Sets and Wikipedia categories based in the server names found and at the end, a view more or less complete depending in the amount of data obtained will be printed.

There is an on-line version of FOCA which can be used by anyone in the following URL: http://www.informatica64.com/FOCA.

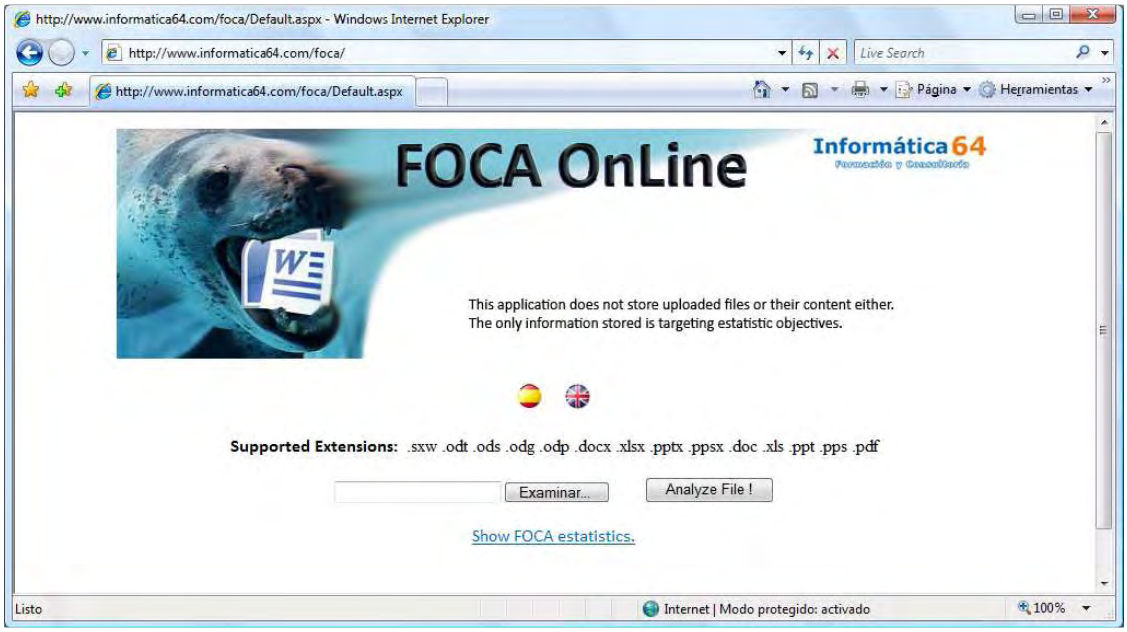


*Figure 49: FOCA, on-line version*

## 7 SECURE WEB PUBLISHING

Removing metadata from documents is not an easy task. As far as users can not be relied on to ensure document security and the volume of published information increases faster and faster, organizations must find ways to make it automatically.

Among the systems involved in document security, web servers play a fundamental role, as they are the logical boundary between Internet and the organization. MetaShield Protector is a solution for sanitizing documents on the fly as they are served to users by IIS web servers. It replaces document metadata so that they contribute to security and offer a normalized public image of the organization.

## CONCLUSIONS

Any document can have associated metadata that contain lost information or hidden data. In particular, Microsoft Office and OpenOffice documents may contain data about the internal network, its user accounts and machines, shared resources, services provided, operating systems, and more.

Organizations must take this information into account before publishing documents on the web or sending them by e-mail. Cleaning these documents is a must, as it is taking care of how web search engines index them.

Keep an eye in the information you know you published and the one you maybe publish without knowing it.

## REFERENCES

EXIF [Exchangeable Image File Format]
http://en.wikipedia.org/wiki/Exif
IPTC [International Press Telecommunications Council]
http://en.wikipedia.org/wiki/IPTC
XMP [Extensible Metadata Platform]
http://en.wikipedia.org/wiki/Extensible_Metadata_Platform
WD97: Cómo minimizar metadatos en documento de Microsoft Word
http://support.microsoft.com/kb/223790
Cómo minimizar metadatos en documentos de Microsoft Word 2000
http://support.microsoft.com/kb/237361
How to minimize metadata in Word 2002
http://support.microsoft.com/default.aspx?scid=kb;EN-US;290945
Cómo minimizar metadatos en Word 2003
http://support.microsoft.com/kb/825576/
How to minimize metadata in Microsoft Excel workbooks
http://support.microsoft.com/default.aspx?scid=kb;EN-US;223789
Ppt97: Cómo minimizar metadatos en presentaciones de Microsoft PowerPoint
http://support.microsoft.com/kb/223793/
PPT2000: How to Minimize Metadata in Microsoft PowerPoint Presentations
http://support.microsoft.com/default.aspx?scid=kb;EN-US;314797
How to minimize the amount of metadata in PowerPoint 2002 presentations
http://support.microsoft.com/kb/314800/EN-US/
Microsoft Word bytes Tony Blair in the butt
http://www.computerbytesman.com/privacy/blair.htm
Word list generation for bruteforce cracking
http://www.reversing.org/node/view/9

### Utilidades

ExifReader
http://www.takenet.or.jp/~ryuuji/minisoft/exifread/english/
Wlgen
http://www.reversing.org/node/view/8
OOMetaExtractor
http://www.codeplex.com/oometaextractor
DocScrubber
http://www.javacoolsoftware.com/docscrubber/index.html
Metadata Extraction Tool
http://www.drewnoakes.com/code/exif/releases/
Libextractor
http://gnunet.org/libextractor/
Bintext
http://www.foundstone.com/us/resources/proddesc/bintext.htm
Metagoofil
http://www.edge-security.com/metagoofil.php
Foca Online
http://www.informatica64.com/foca