Exploitation and Sanitization of Hidden Data in PDF Files

Do Security Agencies Sanitize Their PDF files?

Supriya Adhatarao
Univ. Grenoble Alpes, Inria- France
supriya.adhatarao@inria.fr

Cédric Lauradoux
Univ. Grenoble Alpes, Inria- France
cedric.lauradoux@inria.fr

ABSTRACT

Organizations publish and share more and more electronic documents like PDF files. Unfortunately, most organizations are unaware that these documents can compromise sensitive information like authors names, details on the information system and architecture. All these information can be exploited easily by attackers to footprint and later attack an organization. In this paper, we analyze hidden data found in the PDF files published by an organization. We gathered a corpus of 39664 PDF files published by 75 security agencies from 47 countries. We have been able to measure the quality and quantity of information exposed in these PDF files. It can be effectively used to find weak links in an organization: employees who are running outdated software. We have also measured the adoption of PDF files sanitization by security agencies. We identified only 7 security agencies which sanitize few of their PDF files before publishing. Unfortunately, we were still able to find sensitive information within 65% of these sanitized PDF files. Some agencies are using weak sanitization techniques: it requires to remove all the hidden sensitive information from the file and not just to remove the data at the surface. Security agencies need to change their sanitization methods.

KEYWORDS

Footprinting, sanitization, hidden data, metadata, PDF files.

1 INTRODUCTION

Organization footprinting [6] regroups all the techniques used by hackers to collect as much information as possible about their victims. The goal of this reconnaissance is not only to obtain details on the people but also on the infrastructures they use (network, hardware, system...). Footprinting includes techniques like OS fingerprinting [18] or the exploitation of all the documents published by the organization [1, 2] using tools like FOCA[1]. This later technique is particularly attractive for hackers because it is rather inexpensive and effortless. Organizations publish on their websites many Microsoft Office (`doc`, `docx`...) or Portable

Document Format (PDF) files. All these file formats are particularly interesting for a hacker because they include hidden data which describe the authoring process. The impact of hidden data was highlighted during two events that occurred during the Iraq War.

In February 2003, the British government of Tony Blair published on its website a dossier on Iraq's security and intelligence organizations. The dossier was a Microsoft Word file[2]. The file was analyzed by the IT researcher Richard M. Smith[3] who retrieved the revision logs. It was easy to identify the authors and their positions in government from these revision logs. The British government was greatly embarrassed by the information exposed by those hidden information.

In 2005, an incident occurred between American soldiers and Italian Secret Service officers near Baghdad International Airport causing the death of an Italian officer. Multi-National Force-Iraq issued a report on its investigation of the shooting. That report was posted as an unclassified PDF file with classified sensitive data obscured from public view. However, it was discovered that copying and pasting the classified sections revealed the blocked text.

Hidden data in electronic documents matters and they must not be underestimated. Promptly after these two events, the National Security Agency (NSA) acknowledged in [23] that hidden information in documents are a real security threat:

> **Meta-data and Document Properties** - In addition to the visible content of a document, most office tools, such as MS Word, contain substantial hidden information about the document. This information is often as sensitive as the original document, and its presence in downgraded or sanitized documents has historically led to compromise.

Sanitization is the obvious choice to deal with hidden data before sharing or publishing a document. However, most organizations are unaware that they need to sanitize their

---

[1]https://github.com/ElevenPaths/FOCA

[2]Retrieved 02/24/2021 at http://web.archive.org/web/20040329171413/http://www.computerbytesman.com/privacy/blair.doc

[3]Retrieved 02/24/2021 on http://web.archive.org/web/20040113074742/http://www.computerbytesman.com/privacy/blair.htm

files. In this paper, we investigate how the hidden data of PDF files can be exploited and if they are sanitized. We focus on answering two questions:

- What can be done using the hidden data of PDF files?
- Are there any organizations that sanitize their PDF files?

To answer these questions, we have crawled the websites of 75 security agencies of 47 countries and collected 39664 PDF files. For the majority of the files (76%), we were able to recover the authoring process: we identify the PDF producer tool and the Operating System (OS) used by the file's authors. Collecting and analyzing PDF files from the same source over several years can reveal the habits of a given employee. It is possible to learn if he/she update/change (or not) their software regularly. For instance, we found one employee of a security agency who has never changed or updated his/her software during a period of 5 years. This kind of information is particularly interesting for a hacker to target an individual with bad software habits. By analyzing the PDF files published by several employees of the same agency, it is possible to learn the software policies of an agency. We found at least 19 security agencies in our dataset who are using the same software over a period of 2 years or more. Around 38 security agencies have better practices and are regularly changing or updating their software.

We have also observed that, 24% of all the PDF files have been sanitized. We have identified 7 agencies which sanitize their PDF files before publishing. However, our analysis shows that the sanitization method used was weak for 65% of the sanitized PDF files. It was possible to recover sensitive information from these files. Only 3 agencies are reaching a satisfying sanitization level.

This paper is organized as follows. In Section 2, we present the hidden data included in a PDF file and some related work. The principles of PDF sanitization are given in Section 3. Our results on the sanitization and on hidden data of PDF files are described respectively in Section 4 and in Section 5.

2 HIDDEN DATA IN PDF FILES

Adobe Portable Document Format is one of the most widespread and used file format throughout the world. It has been created by Adobe Systems in 1993 to extend Postscript and it is now standardized as an open format ISO since 2008 and the latest version of the standard is PDF 2.0 [9]. Even though this format has been evolved over time and later versions have been improved to provide better security, it is still not immune to flaws and attacks that could be misused. Many works have been done in the past on PDF file security [4, 12, 20, 22, 26] and privacy [8, 10, 19]. Some works also include results on PDF file sanitization [3, 7, 8].

The visible content of a PDF file can directly reveal many information on the authors: their names, their organization…It can embed many different media: text, pictures, videos…During our analysis of PDF files, we observed that a PDF file is basically collection of indirect objects. There are eight types of objects: boolean, numbers, strings, names, arrays, dictionaries, streams and the null object. Using these objects, data is stored within the PDF file. Sometimes objects in a PDF file can include hidden data that is not displayed while viewing the PDF but can be extracted using appropriate tools.

NSA provides a list of eleven main types of hidden data and embedded content that may be found in PDF files [16, 24]. Only after all these eleven types of hidden information is removed, a PDF file is safe for distribution.

(1) Metadata

(2) Embedded Content and Attached Files

(3) Scripts

(4) Hidden Layers

(5) Embedded Search Index

(6) Stored Interactive Form Data

(7) Reviewing and Commenting

(8) Hidden Page, Image, and Update Data

(9) Obscured Text and Images

(10) PDF Comments (Non-Displayed)

(11) Unreferenced Data

We explain the most commonly found types of hidden data in the rest of this section.

2.1 Dynamic Resources

It was demonstrated in [5] that a PDF can be tracked when it is read. A malicious author can use this technique to obtain information on his/her reviewer during the review process of a conference. It is based on the possibility to embed scripts that are executed by the PDF viewer when the file is displayed. The script needs to download external resources controlled by the author. It exposes the IP address, user-agent and the time of the request of the reviewer. It was even discovered in 2019 that embedding a script was not needed to track a PDF file. CVE-2019-8097 (see https://nvd.nist.gov/vuln/detail/CVE-2019-8097) shows that it is possible to insert an URI in a PDF file which is automatically downloaded by certain PDF viewers. This is the application of web bugs [10, 19, 25] applied to PDF files.

The dangers of scripts and dynamic resources in PDF files are well-known and many solutions exist (see [11, 13, 14, 21, 22]) and the method proposed in [5] is unlikely to work anymore. The exploitation of dynamic resources requires

direct interaction with the targeted organization, *(active footprinting)*. It is out-of-scope of this paper which focuses only on the *passive footprinting.*

2.2   Paths and Images

Images embedded in a PDF file can provide many information on the authors. If an author cleans the metadata of the PDF document but forgets to clean the metadata of the embedded image files, one can still extract information on the author. Image metadata can include author name, username, path from where the file was inserted…Image metadata can include information on creation and modification of the image which can reveal different software used and the habits of the author. Geo-location features stores the geo-data revealing the location where the image was taken which may impacts the security and privacy of an individual.

Several tools like `ImageMagick`[4], `exiftool` and `exiv2`[5] can be used to view metadata. We have used `pdfxplr`[6] in our work to extract the `Username` and `PATH` information from the image files embedded within the PDF documents. Listing 1 shows the `PATH` extracted from one of the PDF files containing a image file. We can see that the author is providing the location where the image file was stored along with the username.

```
1  19693 0 obj
2  <<
3  /K 29/P 19690 0 R/S/InlineShape/Alt(
       Description: C:\\Users\\Mazhar\\Desktop
       \\scml.JPG)/Pg 19761 0 R
4  >>
5  endobj
```

Listing 1: A PDF object displaying the PATH information of an embedded image file.

Image extraction and analysis was proven to be successful in [3, 7] to recover information on the authors of the PDF file. Aura *et al.* [3] have tested 43 anonymous PDF submissions of a conference to detect if any leakage was present. They found 3 submissions which were not properly anonymized. One submission contained a metadata field with the authors names and the two others had image files with identifying metadata. They proposed a tool based on regular expression to detect usernames, device or organization names, identifiers…in a PDF file.

Feng *et al.* [7] have also proposed a tool to detect privacy leakages in PDF files with a focus on the extracted images. The main difference between their tool and the tool of Aura *et al.* [3] is the use of text mining and information retrieval and natural language processing to detect identifying information.

2.3   Comments, annotations and other objects

Comments can be inserted anywhere in the PDF file by starting a line with `%` symbol, PDF viewers just ignore the comments and display the content stored in PDF objects. Comments are often a good source of information.

PDF files sometimes include annotations, these annotations can stay invisible when a PDF file is viewed. However, they can be retrieved using simple string extraction commands like `strings`. Often authors forget to remove the annotations from their files and it can expose their identity. Listing 2 shows an object extracted from a PDF file that reveals the annotated message: `changes made by john doe` and also the Username: `sab`.

```
1  152 1 obj
2  <<
3  /Type /Annot /Rect [165.897704918
       615.5800226116 189.897704918
       639.5800226116 ] /Subtype /Text /M (D
       :20210225232546) /C [1 1 0 ] /Popup 153
       1 R /T (\FE\FF\00s\00a\00b) /P 3 0 R /
       Contents (\FE\FF\00c\00h\00a\00n\00g\00e
       \00s\00 \00m\00a\00d\00e\00 \00b\00y\00
       \00j\00o\00h\00n\00 \00d\00o\00e)
4  >>
5  endobj
```

Listing 2: Annotation object of a PDF file containing the message and username.

Apart from comments and annotations, some specific objects within a PDF file can also include some sensitive information. We found that registry objects, font objects…can include metadata information. Listing 3 shows two objects extracted from PDF files that reveal information about the OS and other software used.

```
1  15 0 obj
2  <<
3  /DW 1000/ CIDSystemInfo
4  <<
5  / Supplement 0/ Registry (Adobe )/ Ordering (
       Identity )
6  >>
7  / Subtype / CIDFontType2 / BaseFont /CAFBBG+
       TimesNewRomanPSMT / Type / Font / FontDescriptor
       23 0 R/W[267[610 443] 284[333]]
8  >>
9  endobj
10
11
12 1459 0 obj
13 <<
14 / Platform (Macintosh )/ Creator ( FileMaker Pro
       Advanced 14.0.1)/ DLI_Copyright ( Datalogics
       Interface \( DLI \) Copyright \(C\) 1998 – 2012
        Datalogics , Inc . -- www . datalogics .com)/
       Producer (Adobe PDF Library 10.1; modified
       using iText 2.1.7 by 1T3XT )/ Title ()/
       Keywords ()/ ModDate (D:20180223153614 + 01 '00 ')
       / Subject ()/ DLI (10.1.0.50)/ Author ()/
       CreationDate (D:20180220152519 + 01 '00 ')
15 >>
16 endobj
```

Listing 3: Specific objects revealing information on the authoring process of a PDF file.

2.4 Metadata

The metadata are often used in cataloging to help searching for documents in external databases. Metadata are stored either in a document information dictionary or in a metadata stream. When the metadata are stored in a document information dictionary, the PDF file's trailer include an optional `Info` entry that holds the document information dictionary consisting of metadata information associated to the PDF file. Metadata information can also be stored in streams called metadata stream. It is either for the complete PDF file or for some components within the PDF file. Metadata stream is represented using Extensible Markup Language (XML).

Many studies in the past have discussed on the impact of PDF metadata [1, 2, 8, 15, 20]. There are several ways and tools that could be used to view and extract metadata information from PDF files: `exiftool`[7], `Metagoofil`[8] and `pdfxplr` to name a few. We have analyzed the metadata of PDF files of our dataset using `exiftool` and `pdfxplr`. We assume that the metadata can be 100% trusted. Some values can be decoys but we cannot detect such situation.

---

[7]https://exiftool.org/

[8]https://tools.kali.org/information-gathering/metagoofil

Table 1 shows the metadata of a PDF file. During our analysis, we noticed that number of metadata fields and their names depend on the PDF producer tool used. Some software also provide an option to the user to either add or remove metadata fields while creating a PDF file.

ExifTool Version Number	: 11.49
File Name	: 127-Zadost-2-10-2018.pdf
Directory	: ./www.vzcr.cz
File Size	: 259 kB
File Modification Date/Time	: 2019:04:04 13:21:11+02:00
File Access Date/Time	: 2020:08:21 13:22:52+02:00
File Inode Change Date/Time	: 2020:07:31 11:23:04+02:00
File Permissions	: rw-r−r−
File Type	: PDF
File Type Extension	: pdf
MIME Type	: application/pdf
PDF Version	: 1.5
Linearized	: No
Page Count	: 1
Language	: cs-CZ
Tagged PDF	: Yes
Title	: MINISTERSTVO OBRANY
Author	: chocholaty
Creator	: Microsoft Word 2010
Create Date	: 2019:04:04 13:16:51+02:00
Modify Date	: 2019:04:04 13:16:51+02:00
Producer	: Microsoft Word 2010

Table 1: Exiftool output for a PDF file in our dataset.

2.5 Positioning of our work

The different types of hidden data found in PDF files are summarized in Table 2. As explained previously in this section, the problem of hidden data in PDF files is well-known. For instance, Mendelman has demonstrated in his master thesis [15] that PDF metadata can be used to footprint an organization. He has gathered 1580 PDF files from 3 organizations and was able to collect printer names, internal domain names, Operating Systems, personal information and producer tool information.

Our work can be distinguished from the previous works by the scale of our analysis. We have gathered PDF files produced by the same authors during several years. It allows us to observe authors and even organization's update policies. Moreover, we have been able to observe how security

Hidden data	Previous Use	Tool
Dynamic resources	[5]	N/A
Path and Images	[3, 7, 8]	`pdfxplr`
Annotations		`strings`
Comments and object		`grep`
Metadata	[1–3, 8, 15]	`exiftool`

Table 2: Hidden data and tools needed to recover them.

agencies sanitize their PDF files. To the best of our knowledge, this is the first study that measures quantitatively and qualitatively sanitization of PDF files.

3 PDF FILES SANITIZATION

In this section, we describe several PDF sanitization tools that we have tested during our work and then we present four different levels of sanitization methods used by authors.

3.1 Sanitization tools

Adobe Acrobat tool: is often mentioned in NSA guidelines [16, 17, 23] as a reliable sanitization tool. It cleans the metadata and all the hidden content of the PDF file. This is the most complete sanitization tool we have used in our work.

GhostScript: Converting PDF file to postscript is mentioned online as one of the sanitization method. This conversion can be done using `GhostScript` tool (Listing 4) and we have observed that it clearly removes a lot of hidden information on the resulting PDF file. However, it is difficult to determine what is removed or kept by this conversion.

```
pdf2ps  filename.pdf
ps2pdf  filename.ps
```

Listing 4: PDF metadata sanitization using Ghostscript.

Exiftool: Several threads on sanitization in online forums also mention the possibility to sanitize a PDF file using `exiftool` (https://exiftool.org/). We observed that, it only cleans the metadata of a PDF file (Listing 5) and not the rest of the hidden information.

```
exiftool -all:all= file.pdf
```

Listing 5: PDF metadata sanitization using exiftool.

Text processing software: can also produce PDF files without any metadata. This is the case for `Microsoft Office Word` or `LibreOffice`.

LaTeXtools: It is also possible to include options in LaTeXsources to remove the metadata (see Listing 6) or use the `pdfprivacy` package[9].

```
\usepackage{hyperref}
\hypersetup{
pdftitle={},
pdfauthor={},
pdfproducer={},
pdfcreator={},
pdfkeywords={},
}
```

Listing 6: Producing PDF file without metadata with hyperref in LaTeX sources.

3.2 Levels of sanitization

During our analysis of PDF files of security agencies, we observed that different authors have used different levels of sanitizations. Hence, we distinguish four different levels of PDF file sanitization:

- **Level-0** consists of PDF files that include complete metadata information. There is no sanitization.
- **Level-1** consists of PDF files with partial metadata. Some metadata fields have been removed.
- **Level-2** consists of PDF files without any metadata. They have been sanitized using `exiftool` or by directly producing PDF files without any metadata.
- **Level-3** consists of PDF files with no information leakage and properly cleaned. All the objects within the PDF file holding sensitive information have been removed. This level can be obtained using Adobe Acrobat tool.

When Level-2 and Level-3 sanitization are observed, we know that the authors of the PDF file have a clear will to sanitize their PDF files. Level-0 and Level-1 are observed when the authors have not applied any sanitization method on their PDF files.

Solution based on `exiftool` and by producing PDF files without metadata (Level-2) are not as strong as Adobe Acrobat with respect to sanitization (Level-3). Therefore, as suggested by NSA, Adobe Acrobat tool should be considered as the only reliable solution for complete PDF file sanitization (Level-3).

4 EXPLOITATION OF HIDDEN DATA

The National Security Agency (NSA) has high security standards and it has provided many guidelines to sanitize PDF files [16, 17, 23]. This was a motivation in our work to check if it was possible to exploit the hidden data in the PDF files published by security agencies around the world. We also

---

[9]https://ctan.org/pkg/pdfprivacy

wanted to know if anybody follow NSA guidelines on PDF sanitization.

We have crawled the websites of 75 security agencies mentioned by Wikipedia[10] belonging to 47 countries. We have downloaded 39664 PDF files in total. The distribution of PDF files over the agency is not even. We found between 5 to around 6000 PDF files for each agency. Figure 1 shows the discrepancy in the number of PDF files for each country in our dataset. We have used `wget` command to crawl websites and download these PDF files.

We have analyzed the content of the PDF files in our dataset to find if the names of the authors appear directly within the document or not. Using text extraction commands, we found the author names in 1783 (4%) PDF files and the rest of the 37881 (96%) PDF files are anonymous.

It is important to notice that we did not know if the authors of the PDF files were some employees of the security agencies or if they were working for agencies contractors. We do not know the ground truth, we have made the assumption that the authors of the PDF files were working for a security agency for simplicity.

In the rest of this section, we describe different information that we were able to extract on the author and on the organization from PDF files.

4.1   Information leaked on the authors

Name of the author: During our analysis of PDF files, we observed that three metadata fields `author`, `creator` and `Tag Author Email Display Name` reveal the name of the author producing the PDF file. In our dataset, 13166 (33%) PDF files reveal the identity of the individual who have created the file.

PDF producer tool used by the author: Metadata fields like `producer`, `creator` and `creator tool` reveal the name of the PDF producer tool used by the author. We found that, in our dataset 30155 (76%) PDF files include the metadata information on the PDF producer tool used. In Table 3, we provide the list of popularly used PDF producer tools. `Acrobat Distiller`, `Microsoft Office Word` and `Adobe PDF Library` are the most popular tools in our dataset.

Producer tool	# PDF	# Agencies
Acrobat Distiller	9054 (23%)	46
Adobe PDF Library	6171 (16%)	50
Microsoft Office Word	4850 (12%)	66
LibreOffice	2171 (5%)	07
Ghostscript	1133 (3%)	36
Mac OS X Quartz	94 (0.2%)	20
SKia/PDF	106 (0.2%)	08
Other tools	16085 (40.5%)	75

Table 3: # of PDF files associated to popular PDF producer tools (76% PDF files).

Operating System used: Interestingly, producer tool name in the metadata sometimes reveals the Operating System (OS) used by the author. The reason is that some tools are OS specific: *Mac OS X 10.6.6 Quartz PDFContext, Acrobat Distiller 20.0 (Macintosh), Acrobat Distiller 8.3.1 (Windows) or Antenna House PDF Output Library 6.2.553 (Linux64)*. In our dataset atleast 16805 (42%) of the PDF files reveal Operating System information. Table 4 shows the distribution of PDF files between the three main Operating Systems: Microsoft Windows, Mac OS and Linux. We can see that Microsoft Windows is a popular choice among the employees of many agencies.

OS used	# agencies	# PDFs
Microsoft Windows	71	11,174 (28%)
Mac OS	29	3,444 (8%)
Linux	7	2,187 (6%)

Table 4: # of agencies and OS used.

Brand of the device, e-mail and Path information: We observed that sometimes authors reveal the brand of their hardware device in the metadata fields author, creator tool and creator instead of their name or along with their name. Using simple string extraction commands, we found four brands like `Toshiba`, `HP`, `DELL` and `Lenovo`. Authors of 24 security agencies have such practices (Table 5).

Using the `pdfxplr tool`, we could also extract different information like the e-mail address of authors, the PATH or location of the folder from where images/files were inserted within the PDF files. We found personal or official e-mail addresses in the PDF metadata. There were 52 unique e-mail addresses in one of the metadata field such as `Tag Author Email`, `author` and `Current User Email`...47 of these e-mail addresses are official that

---
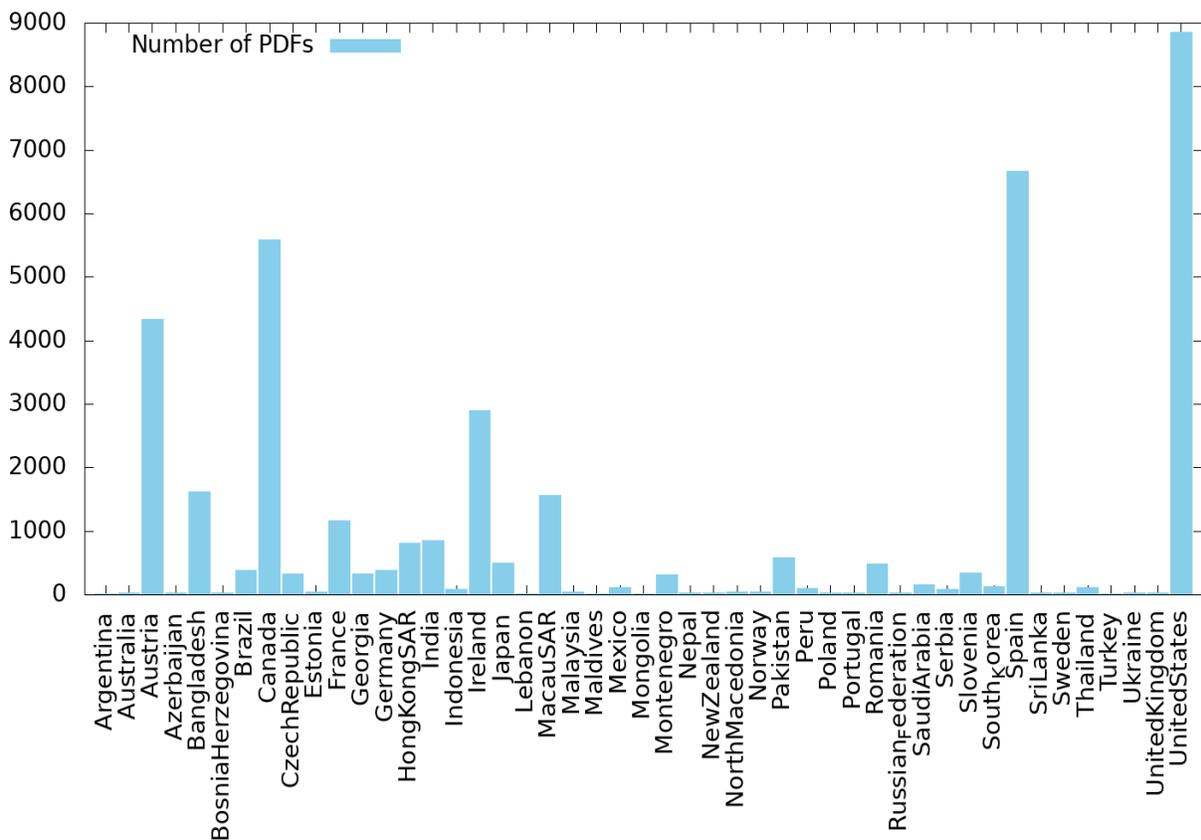[10]https://en.wikipedia.org/wiki/Security_agency

Figure 1: # PDF files for each country in our dataset.

belong to the agency where the author works and four of them are `gmail addresses`, one of them is `outlook address` (Table 5).

OS	# PDF	# Agencies
E-mail	52	13
Hardware brand	581	24
Paths	1814	47

Table 5: # of PDF files revealing e-mail, hardware and PATH information.

Images/files included within the PDF also contain the metadata and the path from where the image was included. Image metadata is out of scope of this paper. We found complete location of where a file is located for 1814 PDF files (Table 5).

Author's behavior w.r.t software updates: *It is possible to combine author, producer and time information provided in*

PDF metadata fields to understand how employees in security agencies update or change their PDF producer tool. We have provided three examples in Table 6. Author-X is working at Federal Investigation Agency (Pakistan) and he/she has never changed/updated his/her PDF producer software from 2014 to 2019. This author is using `Microsoft Office Word 2007` software which is a older version of the software and may contain vulnerabilities that could be exploited. Author-Y works for the Spanish Ministry of Defense and updates his/her `Acrobat Distiller software` on a regular basis. Author-Z is working at the Customs and Excise Department (Hong Kong) and produced PDF files using Adobe Acrobat. Author-Z sometimes used `PDFCreator` tool to convert documents into PDF files on `Microsoft Windows Operating System`.

4.2 Information leakage on the organization

Now we focus on the information that could be exploited on the organization level using PDF metadata. During our

Agency	Author Name	Author habits	Year	PDF producer tool	# PDF published
fia.gov.pk	**Author-X**	**Using same tool**	**2014-19**	**Microsoft Office Word 2007**	**29**
defensa.gob.es	Author-Y	Updating regularly	2010	Acrobat Distiller 7.0.5 (Windows)	4
			2011	Acrobat Distiller 8.0.0 (Windows)	1
			2011-14	Acrobat Distiller 8.2.5 (Windows)	9
			2014-15	Acrobat Distiller 10.1.0 (Windows)	26
			2017-18	Acrobat Distiller 11.0 (Windows)	3
customs.gov.hk	Author-Z	Changing tools	2017	Adobe Acrobat 11.0.20	1
			2018	Adobe Acrobat 11.0.0	1
			2019	PDFCreator 2.1.2.0	3
			2019	PDFCreator 3.2.2.13517	2
			2019-20	Adobe Acrobat Standard 2017 17.11.30150	2

Table 6: Interesting author behaviors observed using PDF producer tool information.

analysis we observed that, many agencies include more than one author publishing the PDF files. It is possible to download all the PDF files published on a security agency's website and observe the author habits, OS trends…

By aggregating the PDF files published for several years, we have considered three different profiles for employees on the organization level (behaviors previously shown in Table 6). **Profile-1** employees update their software on a regular basis. **Profile-2** employees have changed their software. Finally, **Profile-3** employees do not change their software during a period of at least 2 years. Figure 2 shows the number of employees following **Profile-1**, **Profile-2** or **Profile-3** for each security agency. In our dataset, we found atleast 19 agencies that have 154 employees following **Profile-3** and who do not change/update their tools over a period of two years or more.

OS used by agencies: In our dataset, OS details are revealed in 16805 (42%) PDF files. Table 4 shows the distribution of PDF files between the 3 main Operating Systems: Microsoft Windows, Mac OS and Linux and also how many agencies use each of these Operating Systems. It is also possible to spot how many Operating Systems are used in a security agency. We give the example of Austria's Interior Ministry (bmi.gv.at). Figure 3 shows the different Operating Systems we have been able to spot during the last 24 years. In the last five years, the authors of this agency mainly used Microsoft Windows OS to produce their PDF files and fewer PDF files have also been created using MAC OS.
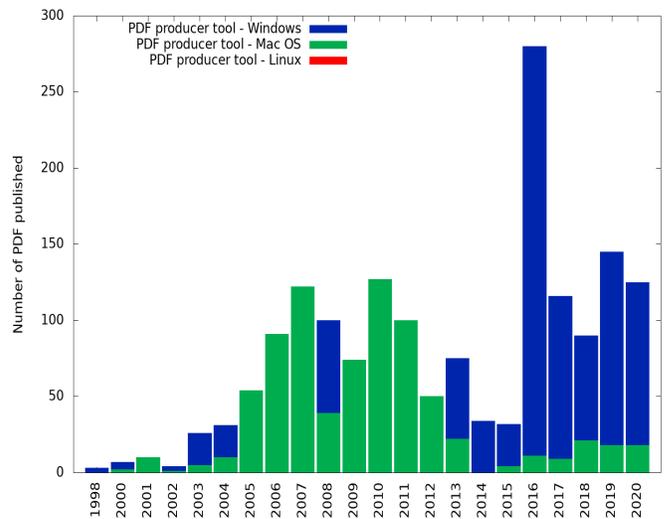

Figure 3: Use of different OS over years at bmi.gv.at.

OS Trends: Using the OS information, we were able to spot if all the employees working for a agency use same Operating System or there exists a diversity and employees use different OSs. Table 7 shows the use of OSs by agencies for a period of 20 years. Even though the number of agencies using same OS is more, we can observe that in the last five years, number of agencies leaning towards using multiple OSs are increasing.
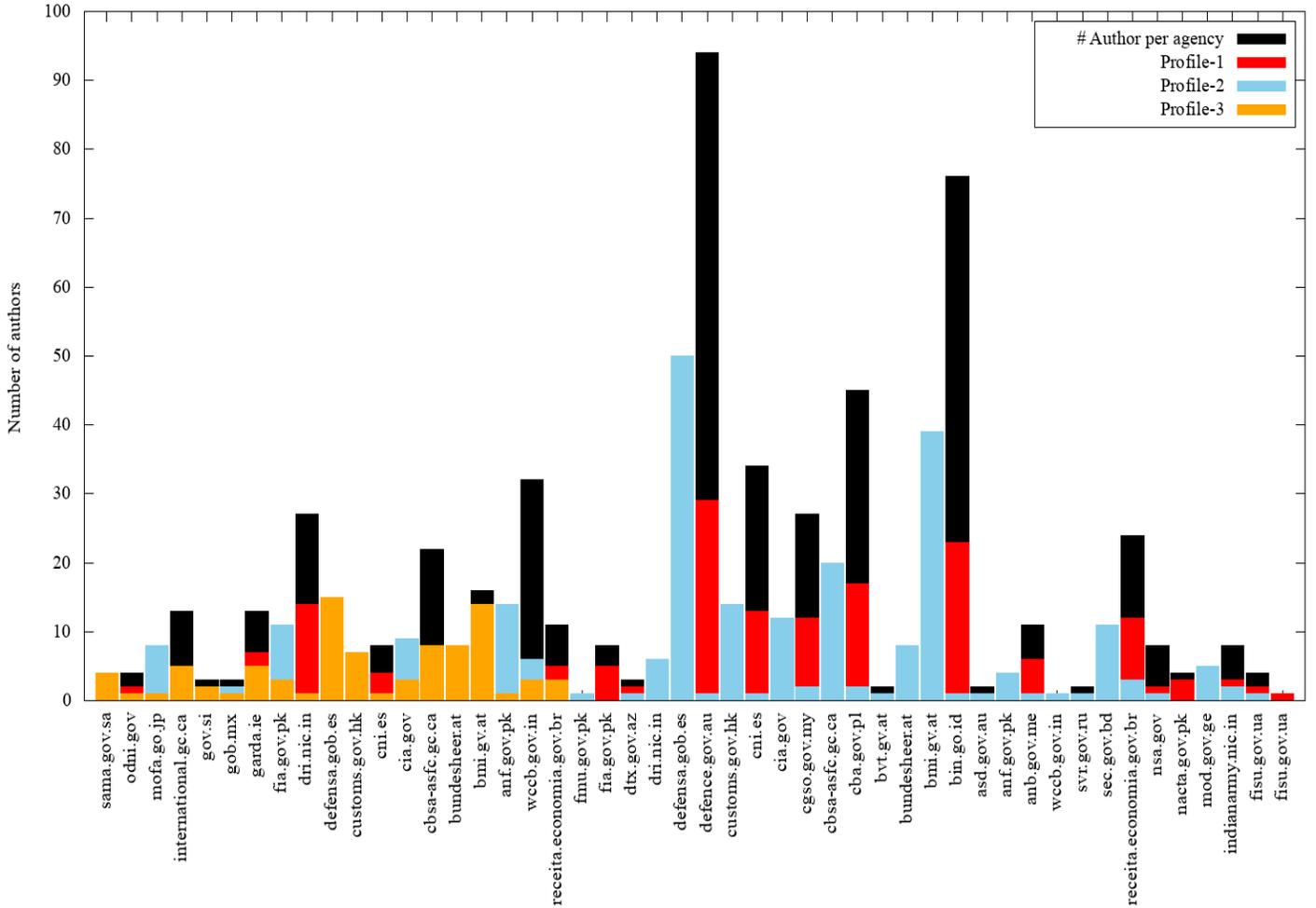
Figure 2: Profile of the employees working in security agencies.

year	# agencies using same OS	# agencies using multiple OS (mix)
2000-2005	15	9
2006-2010	18	10
2011-2015	33	17
2016-2020	37	26

Table 7: # of agencies and OS used.

The results obtained from PDF metadata shows that we can learn different information on the author level and also on the organization level by just using the metadata of the PDF files.

5 SECURITY AGENCIES AND SANITIZATION

We have evaluated each PDF file in our dataset and extracted its metadata to check the level of sanitization. Table 8 provides the number of PDF files for each level of sanitization in our dataset. We found that a total of 9509 (24%) PDF files have been sanitized before being published online. Clearly, PDF sanitization is a concern for several security agencies. However, we found that only 3313 (8%) PDF files in our dataset were sanitized with Level-3.

We have computed a score for each agency based on the level of sanitization of the PDF files published. This score is the weighted sum of the number of PDF files sanitized with a certain level times the corresponding level. The value n_i is the number of PDF files sanitized with Level-i for $0 \leq i \leq 3$

Level of sanitization	# PDF
Level-0 sanitization	16199 (41%)
Level-1 sanitization	13956 (35%)
Level-2 sanitization	6196 (16%)
Level-3 sanitization	3313 (8%)

Table 8: Different levels of sanitization used on PDF files.

(see Equation (1)). The highest possible score is 3 and it can only be achieved if the agency publishes all the PDF files with Level-3 sanitization.

$$\text{Score} = \frac{0 \times n_0 + 1 \times n_1 + 2 \times n_2 + 3 \times n_3}{n}, \quad (1)$$

For instance, we have downloaded $n = 82$ PDF files on nsa.gov agency. We found that $n_0 = 45$ (Level-0), $n_1 = 24$ (Level-1), $n_2 = 13$ (Level-2) and $n_3 = 0$ (Level-3). Therefore, NSA has a score of 0.60.

Table 9 shows the score distribution of the security agencies. One security agency (nabis.police.uk) does not care to sanitize any PDF files before publishing. On the other side of the scale, we found no agency with the perfect score. We found only 7 agencies with a score greater or equal to 2: this implies that most of their PDF files are sanitized. Four of these agencies ssi.gouv.fr, bmi.bund.de, interior.gob.es and secp.gov.pk have performed Level-2 sanitization on most of their PDF files. And atleast, three agencies sie.ro, garda.ie and bvt.gv.at have taken care to sanitize most of their PDF files with Level-3 sanitization.

Score	0	0 >1	1	1 >2	2	2 >3	3
# agencies	1	50	6	11	4	3	0

Table 9: Sanitization score of security agencies.

5.1 Comparative data

In order to put the results obtained for security agencies into perspective, we have also downloaded more than 500000 PDF files from scientific repositories (Cryptology ePrint Archive[11] and HAL[12] with the agreement of the system administrators of these websites.

Are scientists concerned by PDF sanitization? They are at least when they submit their work to a venue which enforce a double-blinded peer review process. Both reviewers

---

[11]https://eprint.iacr.org/
[12]https://hal.archives-ouvertes.fr/

and authors are anonymous from each other throughout the review. Many conferences (47 in computer science according to http://double-blind.org) have enforced a double-blind review submission process. 7 conferences in 2020 (ACM - ASPLOS 2020, HPCA 2020, ICSE 2020, ISCA 2020, MICRO 2019, SIGMOD 2020 and CHI 2020) mentioned explicitly that the authors must remove all identifiable information from the PDF file. ICSE conference also provides instructions to check the metadata using `pdfinfo` or Adobe Acrobat. But none of these 7 conferences provide instructions on how to clean metadata and sanitize the PDF file before submission. Among the 500000 PDF files downloaded, we found only one PDF file that has been sanitized (level-2). *Clearly, security agencies are more concerned by PDF files sanitization than scientists.*

5.2 Why level-2 sanitization is not enough?

As previously mentioned in Section 3, Level-2 sanitization is done either using `exiftool` or by directly producing the PDF without any metadata information. We observed that, if `exiftool` is used to sanitize a PDF file, it is still possible to recover all the metadata.

Metadata are stored in a separate object within a PDF file and `exiftool` only removes the reference to this metadata object in the file. Hence, it is still possible to access this object. Accessing each field of the metadata requires only the use of the `grep` command. In our dataset, we found 1237 (3%) PDF files sanitized using `exiftool` and we were able to recover metadata information using `grep` command for all these 1237 (3%) PDF files.

We also found that 4959 (13%) PDF files have been produced without any metadata. This sanitization method is more efficient but we succeed to extract some sensitive information hidden within the PDF. During our analysis of PDF files, we observed that apart from metadata object, some other objects present in PDF can include hidden information. Using the `grep` command, we performed simple search on the PDF files and found hidden information on the software used by the author producing the PDF file.

Security agencies are recommended to follow proper sanitization methods to avoid leakage of any kind of information within their PDF files.

6 CONCLUSION

Our study shows that the PDF files published by different security agencies are not sanitized to the level expected by such organizations. Many PDF files published by these agencies contained hidden information which can be used to target their employees. We were even able to find an employee who

has not updated his/her software during 5 years. Footprinting an organization using its published PDF files is quite effective.

Some agencies care about sanitization but only 3 agencies out of 7 are doing it properly. The issue is that popular PDF producer tools are keeping **metadata by default** with many other information while creating a PDF file. They provide no option for sanitization or it can only be achieved by following a complex procedure. Software producing PDF files need to enforce **sanitization by default**. The user should be able to add metadata only as an option.

In particular, our study urges for security agencies to take measures that should enforce stronger sanitization methods to limit the risk of information leak in their PDF files.

7 ACKNOWLEDGMENTS

REFERENCES

[1] Chema Alonso and José Palzon. 2009. Tactical Fingerprinting using Foca. In *DEF CON 17 Hacking Conference*. 41–50.

[2] Chema Alonso, Enrique Rando, Francisco Oca, and Antonio Guzmán. 2009. Disclosing Private Information from Metadata, hidden info and lost data. In *Black Hat Europe 2009*.

[3] Tuomas Aura, Thomas A. Kuhn, and Michael Roe. 2006. Scanning electronic documents for personally identifiable information. In *Proceedings of the 2006 ACM Workshop on Privacy in the Electronic Society, WPES 2006*, Ari Juels and Marianne Winslett (Eds.). ACM, 41–50.

[4] Curtis Carmony, Xunchao Hu, Heng Yin, Abhishek Vasisht Bhaskar, and Mu Zhang. 2016. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016*. The Internet Society.

[5] Aniello Castiglione, Alfredo De Santis, and Claudio Soriente. 2010. Security and privacy issues in the Portable Document Format. *Journal of Systems and Software* 83, 10 (2010), 1813–1822.

[6] Jeremy Faircloth. Chapter 2 - Reconnaissance. In *Penetration Tester's Open Source Toolkit (Fourth Edition)* (fourth edition ed.), Jeremy Faircloth (Ed.). Syngress.

[7] Yun Feng, Baoxu Liu, Xiang Cui, Chaoge Liu, Xuebin Kang, and Junwei Su. 2018. A Systematic Method on PDF Privacy Leakage Issues. In *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 12th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2018*. IEEE, 1020–1029.

[8] S. L. Garfinkel. 2014. Leaking Sensitive Information in Complex Document Files–and How to Prevent It. *IEEE Security Privacy* 12, 1 (January 2014), 20–27.

[9] ISO. 2008. *Document management—Portable document format—Part 2: PDF 2.0*. ISO ISO 32000-2:2017. International Organization for Standardization, Geneva, Switzerland.

[10] David M. Martin Jr., Hailin Wu, and Adil Alsaid. 2003. Hidden surveillance by Web sites: Web bugs in contemporary use. *Commun. ACM* 46, 12 (2003), 258–264.

[11] Pavel Laskov and Nedim Srndic. 2011. Static detection of malicious JavaScript-bearing PDF documents. In *Twenty-Seventh Annual Computer Security Applications Conference, ACSAC 2011*. ACM, 373–382.

[12] Davide Maiorca and Battista Biggio. 2019. Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. *IEEE Security & Privacy* 17, 1 (2019), 63–71.

[13] Davide Maiorca and Battista Biggio. 2019. Digital Investigation of PDF Files: Unveiling Traces of Embedded Malware. *IEEE Security & Privacy* 17, 1 (2019), 63–71.

[14] Davide Maiorca, Battista Biggio, and Giorgio Giacinto. 2019. Towards Adversarial Malware Detection: Lessons Learned from PDF-based Attacks. *ACM Comput. Surv.* 52, 4 (2019), 78:1–78:36.

[15] Karl Mendelman. 2018. *Fingerprinting an Organization Using Metadata of Public Documents*. Master's thesis. University of Tartu, Estonia.

[16] NSA. 2008. *Hidden Data and Metadata in Adobe PDF Files: Publication Risks and Countermeasures*. Technical Report. National Security Agency.

[17] NSA. 2015. *Redaction of PDF Files Using Adobe Acrobat Professional X*. Technical Report I73-025R-2011. National Security Agency. https://apps.nsa.gov/iaarchive/library/ia-guidance/security-configuration/applications/redaction-of-pdf-files-using-adobe-acrobat-professional-x.cfm.

[18] Zain Shamsi, Ankur Nandwani, Derek Leonard, and Dmitri Loguinov. 2014. Hershel: single-packet os fingerprinting. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '14*. ACM, 195–206.

[19] Richard M. Smith. 2000. Microsoft Word Documents that Phone Home. (August 2000). Retrieve on Internet Archive: Wayback Machine https://web.archive.org/web/20010203194100/http://www.privacyfoundation.org/advisories/advWordBugs.html.

[20] Charles Smutz and Angelos Stavrou. 2012. Malicious PDF detection using metadata and structural features. In *28th Annual Computer Security Applications Conference, ACSAC 2012*. ACM, 239–248.

[21] Nedim Srndic and Pavel Laskov. 2013. Detection of Malicious PDF Files Based on Hierarchical Document Structure. In *20th Annual Network and Distributed System Security Symposium, NDSS 2013*. The Internet Society.

[22] Didier Stevens. 2011. Malicious PDF Documents Explained. *IEEE Security & Privacy* 9, 1 (2011), 80–82.

[23] Systems and Network Attack Center. 2005. *Redacting with Confidence: How to Safely Publish Sanitized Reports Converted From Word to PDF*. Technical Report. National Security Agency.

[24] Unified Cross Domain Capabilities Office. 2011. *Inspection and Sanitization Guidance for Portable Document Format*. Technical Report. National Security Agency.

[25] Tavish Vaidya, Eric Burger, Micah Sherr, and Clay Shields. 2017. Where art thou, Eve? Experiences laying traps for Internet eavesdroppers. In *Workshop on Cyber Security Experimentation and Test, CSET 2017*. USENIX Association.

[26] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016*. The Internet Society.