University of Amsterdam
Master System and Network Engineering
Research Project 2

Unintended metadata leakage at the Dutch Government

Auke Zwaan
auke.zwaan@os3.nl

July 21, 2017

Supervisors:
Alex Stavroulakis & Sukalp Bhople
KPMG

Abstract

After saving a file, the metadata that gets saved with it contains an abundance of information. However, it is the question whether many people and organizations realize this when publishing files online...

In this research, more than 450,000 public documents from Dutch government domains were searched for, downloaded and processed. After extracting the metadata from all of them, it was shown that the organizations behind the domains can be mapped out using just these data. Based on the information on creation and modification of documents stored in metadata, social networks were drawn and lists of phishing targets were established using simple graph theory. Furthermore, behavioral pattern analysis of both organizations as a whole and specific users was successfully performed. Additionally, thousands of file paths were found, allowing one to get insight into an organization's internal file systems, their structures and eventually even the local network. Finally, vulnerabilities could be found, albeit with some limitations.

Acknowledgments

Apart from any staff, supervisors and others who helped me during this research project specifically, I would like to thank Liz Vink, who gave me unconditional support throughout the years and without whom I would never have been able to do this study.

Contents

1 Introduction

In late 2012, John McAfee was running from law enforcement in Belize. For a while he was untraceable. However, things changed after he talked to Vice Magazine on December 3 that year, which posted a photo of the secret encounter[19][9]. As it turned out, Vice had not reviewed the photo's metadata before publishing it. In fact, they had unknowingly revealed the exact coordinates of the fugitive, who turned out to have fled to Guatemala. While this is about an individual, it is a striking case of how metadata in public files can result in a data leakage of some kind.

For an organization, there might be more at stake. After the hacked email database of the Italian company Hacking Team[1] leaked in July 2015, researchers showed how the metadata from just those emails[2] could be used to paint a picture of the daily life inside the organization[13]. Using usernames and timestamps, for example, interrelations between people could be found, therewith enabling the researchers to map out a social network. For a malicious actor, this information can be beneficial in the reconnaissance phase of a cyber attack. Obviously, it is in the interest of the organizations in question to prevent this from happening.

As the Internet grows in size, more information will inevitably be exposed to the public. As an example, a quick Google search shows that, in June 2017, Google had indexed that more than 30 million Portable Document Format (PDF) files on domains under the *.gov* top-level domain (TLD) alone[3], and more than 1 million Word documents (*.doc* and *.docx*)[4]. This shows the scale of the matter, and the importance to find out more about the current state of affairs. The point here is that an organization might already be exposing information unknowingly, without having been compromised like Hacking Team. In general, metadata in public documents can contain an abundance of information about an organization, such as details about the creation and location of files, usernames, software and hardware, and possibly even more.

This research focuses on the Dutch government specifically, as it is carried out for the University of Amsterdam. The goal is to examine what kinds of metadata from Dutch governmental and semi-governmental websites, such as those of ministries and municipalities, can be found online by parsing documents published on their websites. After gathering that information from hundreds of (semi-)governmental websites, it is analyzed and quantified. Eventually, it is also reviewed whether it is possible to "enrich" the discovered data to make it more meaningful and to create a better picture of the domains in question.

Eventually, the tools and research will be made available so that it can be used by organizations to get insight into and to prevent abuse of metadata leakage on their domains.

2 Research question

One main question is used for guidance throughout the research:

> **What types of metadata does the Dutch government leak through public documents, and how could this information be used in the reconnaissance phase of a cyber attack?**

In order to answer this question, some sub questions will be addressed and answered:

1. What types of documents are interesting to harvest metadata from, and why?
2. What types of metadata found in those documents are interesting for an attacker?
3. How can the information be analyzed, quantified and enriched?
4. How can this information be used for finding potential attack vectors?

[1] http://www.hackingteam.it/
[2] https://wikileaks.org/hackingteam/emails/
[3] https://www.google.nl/search?q=site:*.gov+filetype:pdf
[4] https://www.google.nl/search?q=site:*.gov+filetype:doc

3 Related research

The severity of metadata leakage is underlined by the SANS Institute[5] in an article explaining what types of files (can) contain what types of metadata, and how this reveals information[18]. Another paper on "hidden" information was published by Blackhat[4]. Besides showing how to find the data, it also explains how to remove it from files. This might be useful for organizations who want to prevent any metadata leakage.

In the past, multiple tools focused on harvesting and sometimes also analyzing metadata. Possibly the best example is Fingerprinting Organizations with Collected Archives (FOCA)[6], which can be used to automatically scan a given domain for public documents. Afterwards, it is automatically analyzed and displayed in a Graphical User Interface (GUI). While this is useful for a single domain, feeding it a list of domains from an external script is practically impossible as it can only be used in its GUI form. Furthermore, it only works in a Windows environment, which makes it unable to dynamically communicate with common Linux tools and utilities.

As an alternative, *metagoofil*[7] was launched in 2011. Natively supporting Linux, it used to be able to extract public documents like FOCA would, but at the moment of writing, it no longer works. In fact, it has not been updated for years and is no longer supported by the authors. Another tool, called *Goolag Scanner*, was launched by hacking collective *Cult of the Dead Cow* in 2008. As this tool was launched more than nine years ago, it is not functional on the current version of the Google search engine anymore, and has for long been out of support.

In the same year, a paper on methods for automated metadata extraction was published [15]. While this research focused on extracting metadata from disk images in a forensic setting (and not from public places online), some of the techniques for processing the files themselves might still apply.

Lastly, *Maltego* can be used to visualize connections between entities. However, it is closed source and the free version, *Maltego Community Edition (CE)*, only returns 12 entities per transformation[8]. In Maltego, a transformation is an action applied to an entity (which can be a domain). That means that, at most, 12 documents can be retrieved for a specific domain. As a quick initial search already showed that many of the websites to be reviewed host tens of thousands of public documents, this does not seem like a usable tool for this matter. Furthermore, the most expensive paid version, *Maltego XL*, can only return 64,000 documents per domain, which might not be enough in some cases either. Another version, *Maltego Classic*, costs 360 US dollars per year (with an additional initial cost of 760 US dollars), and it can only handle 10,000 entities per graph[9]. All in all, *Maltego* does not seem to offer the features required for this research either. This means custom tools will have to be written to gather the data. For the data analysis part, regular quantification methods and Python scripts should be suitable.

4 Approach and Methods

The research consists of three main parts: **harvesting URLs of public documents**, **downloading the documents** and **analyzing and visualizing the documents**. In this section, the steps taken to get to the results are described. The according results follow in Section 5.

4.1 Defining in-scope domains

Before being able to follow the steps described in this section, it is important to define a clear scope of what the term "Dutch government domains" encompasses.

Since 2016, the Open State Foundation[10] gives real-time insight into the current level of SSL-adoption amongst Dutch governmental websites through a public live dashboard. While the SSL part of their project was not of any use for this research, the free dataset of government websites (which is available in CSV format) was. As described on the information page of *Pulse*, the title of

[5] https://www.sans.org/
[6] https://www.elevenpaths.com/labstools/foca/index.html
[7] https://code.google.com/archive/p/metagoofil/
[8] https://www.paterva.com/web7/buy/maltego-clients/maltego-ce.php
[9] https://www.paterva.com/web7/buy/maltego-clients.php
[10] https://openstate.eu/

the project, the list of over 1,600 websites in the dataset are gathered from sources directly from the government[17]. It was thus considered a good dataset to use in this research as well.

4.2 Step 1: Identifying public documents

First, a list of interesting common file types was made. This was to prevent unnecessary workload and to narrow down the scope of the research. Initially, the following file types were searched for:

- Microsoft Word documents (**.doc**,**.docx)**
- Microsoft Excel documents (**.xls**,**.xlsx**)
- Microsoft Powerpoint documents (**.ppt**, .**pptx**)
- PDF documents (**.pdf**)
- Apache OpenOffice documents (**.ods**, **.odt**, **.odp**)

This list was established after an initial, short analysis of common files and what information is in their metadata. As it seemed, the Microsoft Office documents contain relatively much information. They were, therefore, considered relevant documents. Their counterpart, the *Apache OpenOffice*[11] documents were also included for this reason. As PDF documents are also a common file type of published documents, it was logical to include it in the search as well.

4.2.1 Passive scanning

Using the list of file types mentioned above, a "passive scan" was first performed. In this definition, a passive scan means querying a third-party website for any information it has on public documents hosted at the reviewed domains.

Both Google and Bing allow one to use so-called *dorks*, such as "*site:os3.nl filetype:pdf*"[12]. This specific dork shows all indexed documents on the website `os3.nl` that have PDF as their file type. By gently asking Bing for the results for search queries like this multiple times, this task could theoretically be automated. A first look at both Google and Bing showed that only a maximum of around 1,000 results could be browsed through this way, which meant it was not possible to see **all** results after doing the query mentioned above. Although 1,000 results were not enough for many domains, it was still tested whether this task could be automated and used if stealthiness is required.

4.2.2 Active scanning

To increase the number of public documents found, an active scan was performed, too. In this case, a custom crawler that specifically searched for file types from the file type list was used. Starting at the root of the domain, it saved all occurrences of links to interesting public documents to a file. This included all documents with an "interesting" file type, which were in the scope of the domain in question (i.e. external URLs were excluded automatically). Additionally, any rules for crawlers stated in *robots.txt* files were adhered to. This means that there might be parts of the domains that were not crawled. Any documents on such pages were thus not included in this research either.

While public documents linked to from the domain itself appear in the output of the algorithm in this approach, it misses the documents linked to from external domains. To make this clearer, if on **domain A** a document on the same domain is referenced, the crawler finds it. However, any references on **domain B** (out of scope) to **domain A** are not found this way. These incoming links are also called *backlinks*, and right now, there does not seem to be a way to get an overview of them without crawling the entire internet.

For active scanning, the Python framework *Scrapy*[13] was used as it takes *robots.txt* files into account by default and offers easy ways to configure other settings. To limit any disturbance at the domains to be crawled, a delay of 0.1 second was set between each request. This was considered appropriate as Google reportedly does the same [2][3][1] and this research had to be carried out within one month (i.e. if a lower crawling speed would be used, it might have resulted in the results not being there in time). Furthermore, the email address of the researcher and the goal of the research was put into the user agent string that went with every request.

[11] `https://www.openoffice.org`
[12] `https://www.google.nl/search?q=site:os3.nlfiletype:pdf`
[13] `https://scrapy.org/`

One last thing to mention here is that only looking at file extensions will not find all documents. If a website uses rules to hide specific file types (which is common, for example, for *.php* pages, but can be done for any file type), there will not be an extension at all. In these cases, the *Content-Type* response header can show the file type of the returned page. This was thus taken into account when building the crawler, too.

4.2.3 Storing the output

As the custom crawler cannot be used to find backlinks, it is probably outperformed by Google and Bing in certain cases. To compensate for that, the active and passive scans were planned to be combined to get more unique results. The plan was to store the output of both scans, and to then concatenate and convert them into a list of unique URLs linking to public documents.

However, as Bing decided to block the IP address the research is performed from, this part had to be excluded from the research as it was not considered ethical to try and trick search engines into thinking the passive scanning tool is a human by all means.

Note that in this step, the files were not yet downloaded. Instead, this was part of the second step.

4.3 Step 2: Processing the documents

After performing the scans, the locations of the public documents were known. Using a simple loop, the documents were downloaded with a short delay in between to prevent the target servers from becoming overloaded.

To spread the workload even more evenly over the domains, a custom download "pipeline" was used. What this means is that, instead of downloading all files from a domain sequentially, and then proceeding to the next domain, the first document from each domain was first downloaded. Then, the second document was downloaded from each domain. For this reason, the download delays did not have to be long, as the script looped through different domains for every download automatically. Only at the end of the pipeline, it happened that there was only one domain left with files to download. This happens if there are, for example, 20,000 files on one domain, but only 15,000 on the rest. For the last 5,000 rounds of downloads in the pipeline, then, files are only still downloaded from this one last domain. In these cases, a delay was still necessary.

Lastly, the analysis of metadata using *exiftool* took roughly 200 milliseconds, which causes a "natural" rate limit of around 5 requests per second. This was considered appropriate.

4.4 Step 3: Analyzing the metadata

After the documents were downloaded for a specific domain, the files were analyzed. Each file was first downloaded (using the Python *Requests module*[14]), after which it was immediately processed using *exiftool*[15]. For this step, too, a user agent identifying the researcher, and the goal of the research, was provided so that administrators of domains could get in contact in case of any issues. After the metadata had been extracted, the original file was overwritten with the metadata found in that file. At the end of the loop (i.e. after processing all the files), the working directory thus only had files containing the metadata of the downloaded files, and not the contents of the files themselves anymore.

As different files and file types have different metadata *tags*, no distinct selection was made beforehand. As the output of *exiftool* is plaintext only, it could easily be stored and searched through later at any moment.

After gathering all metadata, regular expressions were used to search for details on users, software, pieces of equipment used, dates and times. It was then reviewed whether the data could be further enriched. Also, potential vulnerabilities were researched. For example, an out-of-date piece of software, a specific piece of hardware with known vulnerabilities or an original file location that gives insight into the internal file systems might reveal a weak spot in an organization. It was also reviewed whether any discovered usernames could be linked to real persons in some way as this would enable an attacker to acquire new attack vectors, such as spearphishing.

Eventually, the current state of affairs at the Dutch government when it comes to metadata leakage was evaluated and a conclusion was drawn.

[14] http://docs.python-requests.org/en/master/
[15] http://www.sno.phy.queensu.ca/~phil/exiftool/

4.5 Detailed technical steps (scripts)

In this subsection, the workflow of the custom scripts written for this research is briefly explained. These scripts are to be made open source (if they are stable), and can then be found on https://github.com/AukeZwaan/metadata-tools.

1. **Get a list of websites**
   First, a list of websites must be provided and stored in a file. This file must contain a full URL of a domain to be crawled (including the protocol (*HTTP*/*HTTPS*)).

2. **Crawl the websites**
   Now, use the crawler to find all files of interesting file types on that domain (specified in the source code of the crawler).

3. **Download all files, and analyze with *exiftool***
   Download all files using *downloader.py*. This script will automatically also analyze the files as it processes them, and store their metadata in the output directory.

4. **Create a list of all analyzed files**
   The next step is to create a list of all analyzed files (using *bash*, for example. This list has to be saved as **all_files.txt** as this is the file name the metadata analysis tools will look for.

5. **Run *calculate_centrality.py***
   This script analyzes information on users and who worked together with whom. It creates **nodes.csv** and is used for the **phishing_targets.php** page in the metadata dashboard. It also creates **triadic_closures.csv**.

6. **Run *domain_info.py***
   This script gathers and summarizes information for each domain. It stores the output in **domains.json**.

7. **Run *dir-finder.py***
   This script tries to find all Windows file paths listed anywhere in the metadata of documents using regular expressions. It stores the output in **all_domains_with_files.csv** and **data/<domain>/filepaths.json**, which are later used by the dashboard for the file system visualizations.

8. **Run *time_grabber.py***
   This script analyzes temporal information from the *Create Date* and *Modify Date* metadata tags. It stores the output in **create_and_modify_dates.csv**, which is later used for the timeline visualizations in the dashboard.

9. **Initialize the dashboard**
   Make sure a working web server is running on the local machine. Next, download the metadata dashboard from https://github.com/AukeZwaan/metadata-dashboard and place it in the web directory. If configured properly, it should automatically (graphically) show all results of the metadata analysis. Investigators can now interpret those results.

5 Results

5.1 Harvesting and downloading the documents

While performing the active scan to retrieve the locations of public documents, some domains turned out not to be able to handle around four to five requests per second on some pages. This was largely caused by the fact that those pages rendered web pages dynamically. This means that, for each request, the backend systems generated the pages requested. When the scanner crawled those computationally heavy pages, they faced trouble. This resulted in some server-side blocks towards the scanner. In two cases, the administrator of a domain being scanned kindly asked to reduce the number of requests for this reason. To avoid any further disturbance (and considering the dataset was growing to a reasonable size anyway), those domains were not crawled any further. Due to the immaturity of the scanner at the moment of research, it was decided to remove any hospital domains from the domain scope for ethical reasons, too. Due to time constraints, domains of the category "Gemeenschappelijke Regelingen" (websites of governmental bodies working together) were also left out.

Limits to passive scanning It turned out to be really hard not to be detected as a bot by both Google and Bing. While a script doing Bing dorks did actually work for a couple of days, this was not feasible through Google. After a couple of days, however, the IP address was blocked by Bing already and the passive scan had to be left out.

Processing the documents For processing the documents, another script was used. This script took the list of URLs of public documents as its input and then downloaded each file. After a file was downloaded, before proceeding, it was parsed for metadata using *exiftool*. Then, the original file was replaced by the exif data only, as the available storage was limited and the exact contents of the files were considered out of scope. Domains at which the IP address of the passive scanner was blocked for any reason were skipped here.

In total, files from 471,695 unique URLs were downloaded this way. For 14,462 unique URLs, a 404 HTTP response code was returned. Due to parsing errors for some files, 11,951 could not be parsed for metadata using *exiftool*. This leaves 459,744 files which were successfully downloaded and processed. These files were found on 675 unique domains (there existed domains, which were either really small or which the active scanner had trouble going through; see Appendix A). This comes down to an average of 685 files per domain. For a complete list of all domains on which files were successfully downloaded and processed, see Appendix D.

5.2 Metadata statistics

Often, indicators of software usage could be found at multiple tags within one document. As an example, Table 1 shows the top 10 of tags where the string "Microsoft" was was found, and how many times it was found there. In most of the cases, the term "Microsoft" is used in conjunction with a software version string of a "Microsoft Office" product like *Word*, *Powerpoint* or *Excel*, followed by a year.

The "Primary Platform" tag seems to show the operating system of the machine the document was created on. The only three values found there were **Microsoft Corporation** (1715 times), **Apple Computer Inc.** (300 times) and **Sun Microsystems Inc.** (10 times). Out of more than 450,000 files, these are not high numbers.

Tag	Occurrences
Producer	102,083
Creator	86,343
Title	58,598
Creator Tool	48,423
Software	12,429
Comp Obj User Type	7,795
Application	2,645
Primary Platform	1,715
Generator	964
Author	239

Table 1: Metadata tags where the string "Microsoft" was found (case-insensitive).

When running *exiftool* to extract any metadata from a file, some additional metadata tags are generated. These can be found in Table 2 and will be excluded in any other analysis as they were present in almost all files. After leaving them out, a top 10 of most popular metadata tags and their number of occurrences was created. This top 10 can be found in Table 3.

Tag
File Size
File Permissions
File Modification Date/Time
File Inode Change Date/Time
File Access Date/Time
ExifTool Version Number
Directory
MIME Type
File Type Extension
File Type

*Table 2: Metadata tags generated by **exiftool** which were thus present for almost all files.*

Tag	Occurrences
Create Date	448,513
PDF Version	440,226
Linearized	440,226
Page Count	438,382
Producer	414,724
Modify Date	408,815
Creator	394,072
Author	302,880
Title	282,592
XMP Toolkit	264,469

Table 3: The top 10 most popular metadata tags found in the entire dataset. Note that the tags listed in Table 2 are excluded here.

5.3 Temporal information and analysis

The *Create Date* and *Modify Date* tags show the exact time a document was created or modified respectively. It turned out patterns were visible within these dates, through which, for example, weekends and public holidays could successfully be identified.

Figure 1 shows how many documents were created at what time in the year between 2007 and 2017 (the last 10 years at the moment of writing). It is immediately visible that there are peaks around May each year. Closer analysis showed that this was due to one specific domain publishing high school exams after they are taken (which is at the end of May each year). In Figure 2, this domain is excluded. This shows an entirely different timeline. While there is an increase in the number of documents created each year, the months July and August still seem to show a clear dip. This is most likely due to public holidays. The timelines for the modification dates of documents shows a similar picture.

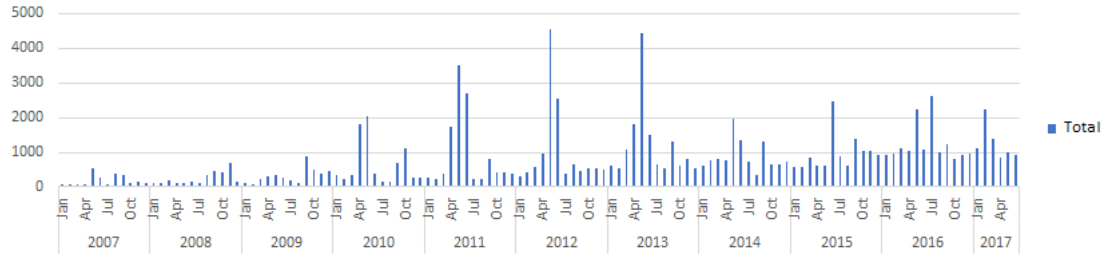

Figure 1: The distribution of the "Create Date" tag found in public documents on all domains. May in every year shows a clear peak.

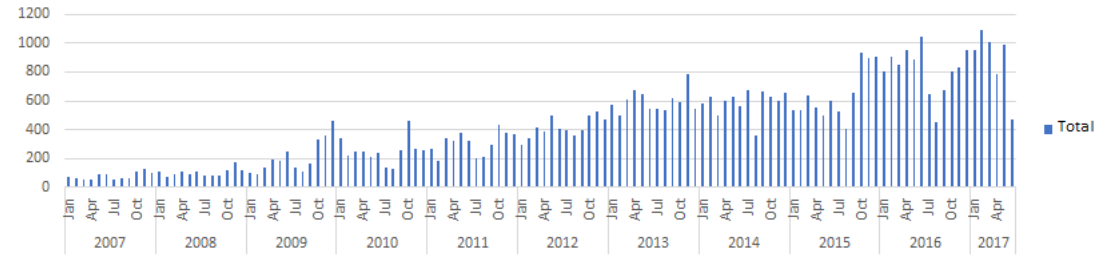

Figure 2: The distribution of the "Create Date" tag found in public documents on all domains, excluding one domain publishing high school exams each year around May. Without this domain, there are no apparent peaks. July/August show a dip each year, which is probably due to public holidays.

In Figures 3 and 4, the creation and modification times of documents are plotted per hour. Clearly, there is a dip between the values 11 and 13, while most Dutch companies have lunch at around 12:00. What caused the spike at 22:00 for both graphs is not clear, however. This can be due to an outlier (which was also the case in Figure 1), but it can as well be caused by an error in the time parsing at the time of analysis, or even people creating and modifying documents from home after dinner. It seems best to get to a conclusion about this per domain, while this can be specific to one organization. It is clear, however, that working times of a domain can effectively be plotted this way. For instance, the organizations analyzed in this research show that, generally, people start working between 7:00 and 8:00, and go home between 17:00 and 18:00 with a break between 12:00 and 13:00.

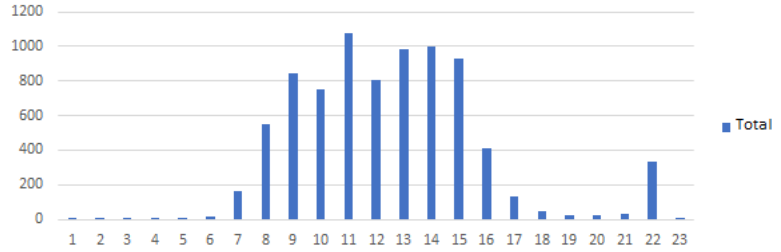

Figure 3: The creation of files per hour.

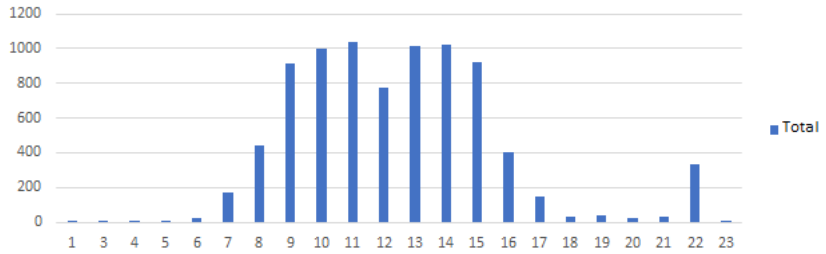

Figure 4: The modification of files per hour.

The activity on different days of the week seems to be distributed relatively evenly, Tuesday and Thursday being the most "productive" days. What is interesting to see here is that Friday is the least productive working day. This is likely caused by the fact that a full working week at the Dutch government is 36 hours (not 40)[16], which means employees often choose to work less on Fridays. This assumption is underlined by Figure 7, which shows a strong drop in productivity (i.e. the number of documents created per hour) from 14:00 on. In fact, the productivity does not come back at its original level after the lunch break, which was the case in Figures 3 and 4.

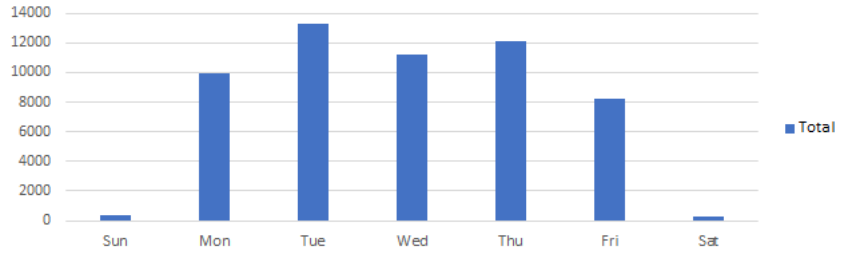

Figure 5: The number of documents created per day of the week.

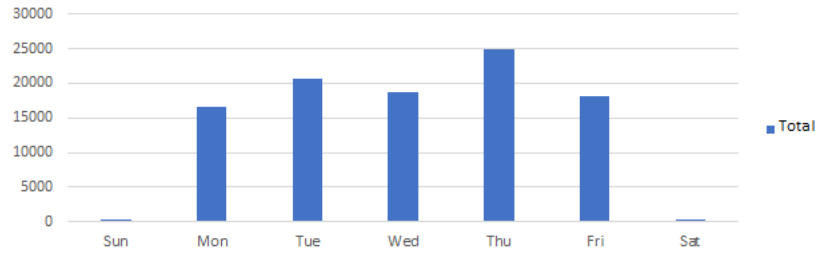

Figure 6: The number of documents modified per day of the week.

[16] https://www.werkenvoornederland.nl/over-de-rijksoverheid/arbeidsvoorwaarden

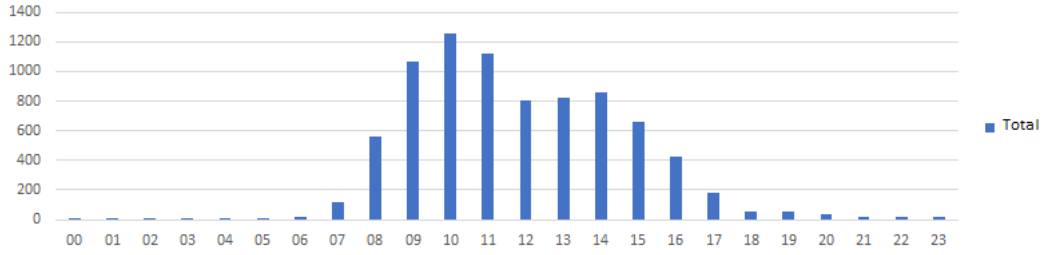

Figure 7: The number of documents created per hour on Friday. The afternoon is significantly less "productive" (i.e. less files are created) than the morning. In fact, the producitivy does not come back at its original level after the lunch break (which is the case for other days, as can be seen in Figures 3 and 4.

The latter analysis can be done for specific users. By looking at the *Creator* tag in conjunction with the *Create Date* tag, plots can be made to see which users worked for a specific organization at a specific time. Days on which people have never created or modified a document could reveal what their "weekly day off" is, for example. This might be interesting for social engineering attacks. Overlapping activity for different users can give one insight in what people might have worked together. In Figure 8, all document creations of a real user from the dataset are plotted over time. Judging on the data, he or she has worked for the Dutch government from November 2009, and still works there. Additionally, it looks like the user went on holiday around July/August each year, which is not uncommon.

In Figure 9, the same user is taken and the hourly productivity (as measured by the number of files created, again) is plotted. As can be seen, the user has never created files before 08:00, normally takes a break between 12:00 and 13:00 and is generally not as productive in the afternoon as he or she is in the morning.

Finally, as mentioned before, the most active days can be plotted for the user, too. Judging on the data displayed in Figure 10, the user likely works less on Fridays. Again, this is based on the assumption that creating less documents on a day equals less productivity, which seems a valid statement.

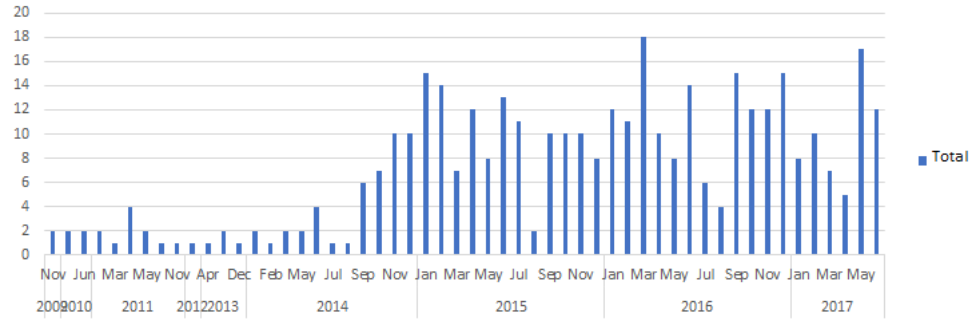

Figure 8: The number of documents created by a real user from the dataset over time. Judging on the data, he or she has worked at the Dutch government from November 2009, and still works there. There are also drops around July/August each year, which might indicate holidays, for example.

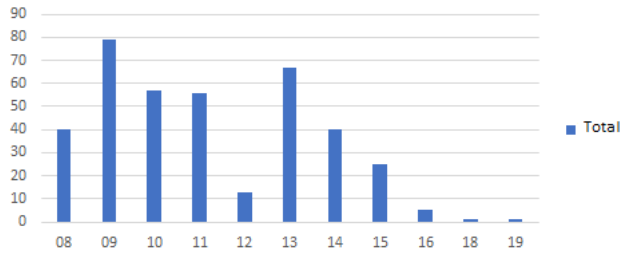

Figure 9: The number of documents created by a real user from the dataset per hour. The user has never created a document before 08:00, normally takes a break between 12:00 and 13:00 and is generally not as productive in the afternoon as he or she is in the morning.

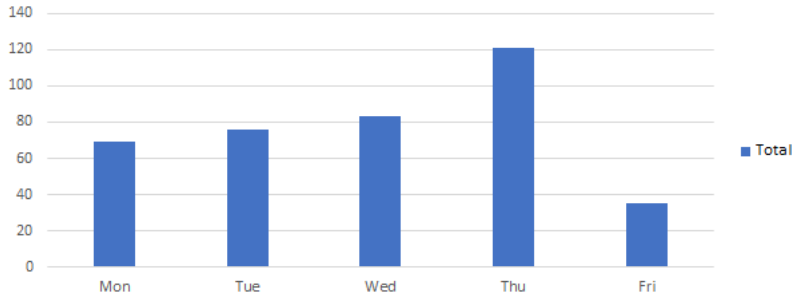

Figure 10: The productivity of a real user per day of the week. It is clear that, on Fridays, the user is much less productive than he or she is on other days in the week.

5.4 Social connections in an edit network

As the Table 3 shows, over 300,000 files have a populated "Author" tag (around 65 percent of all documents), whereas close to 400,000 documents have a populated "Creator" tag (over 85 percent of all documents). This shows the ubiquity of data about users. In this section, it is described how analysis was done based on these data, thereby creating an "edit network".

5.4.1 Implicit and explicit relations between users

In the metadata of Microsoft Office documents (which includes all files with a *.doc*, *.docx*, *.xls*, *.xlsx*, *.ppt*, *.pptx* file type), there are three interesting metadata tags: **Creator**, **Author** and **Last Modified By**. In short, the *Creator* and *Author* tags will mostly contain the username of the username of the original creator of the document. Then, if the document is modified by a user (which can also be the same) at some point in time, the *Last Modified By* tag is populated with that user's username.

If user **A** was the creator of a document **X**, which user **B** later modified, their usernames will thus be present in the *Creator* and *Last Modified By* tags of document *X*, respectively. This also means that they worked together on the document, and that there is at least some relationship between the two users.

Now, if user **A** also created another document, **Y**, which was modified by user **C** at some moment in time, there is also a relationship between user *A* and user *C*. Applying transitivity here, it can be said that user *B* and *C* now have an implicit relationship through user *A*, too. In Figure 11, this example is depicted.

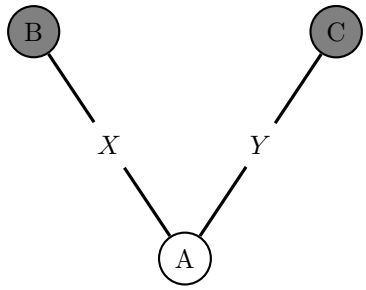

*Figure 11: If user **A** worked together on document **X** with user **B**, and on document **Y** with user **C**, there is an implicit relationship between user **B** and **C** through user **A**.*

Building further on the inferred relationships between users who edited a document together, larger **edit networks** can be constructed. A real-life example, derived from the metadata found in Microsoft Office documents on a domain in the dataset, can be seen in Figure 12.

In total, after processing all domains in the dataset, 30,978 unique pairs of users subject to triadic closure could be found this way.

When applying this theory, it is important to keep in mind that the *Last Modified By* tag will only contain the **last** modifier of the document (i.e. not a list of all modifiers). This means that there might be a situation in which another user edited a document in between the two listed users. This user cannot be retrieved from the information present in the metadata.

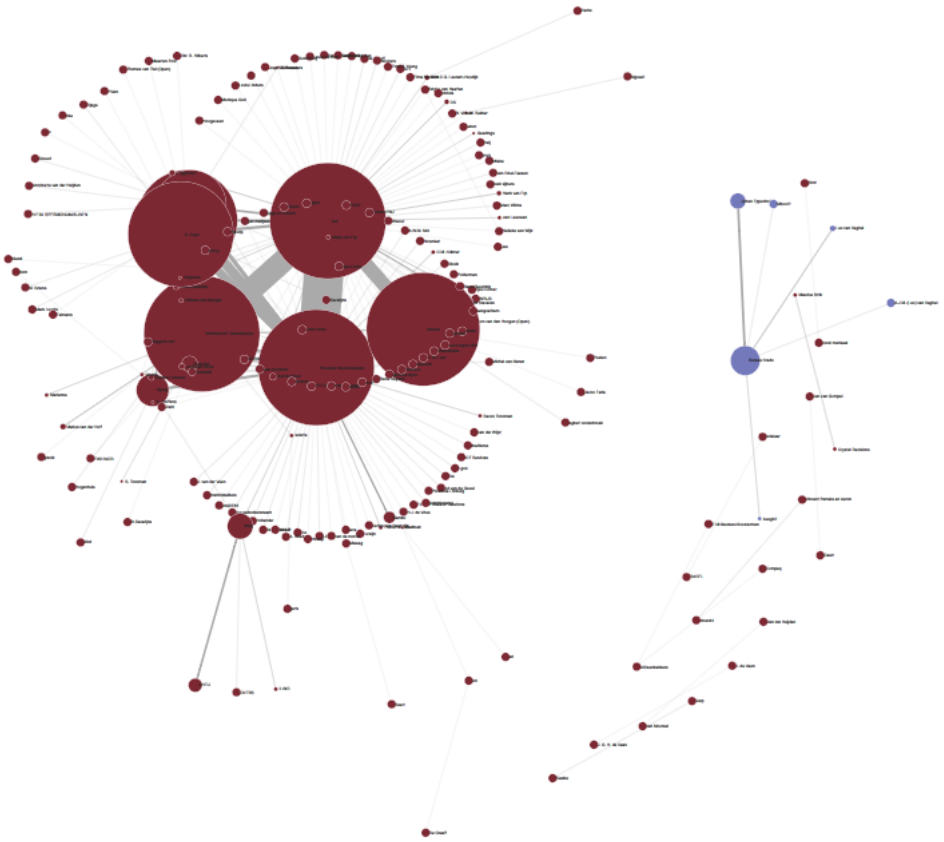

Figure 12: A real-life example of the edit network in a domain from the dataset. While the exact usernames are not important for this figure, it is clear that there are many connections between the users on the domain. Larger nodes depict more documents in which that user was found as the creator or the modifier, and thicker edges mean two users worked together on documents more times.

5.4.2 Triadic closure

After creating the edit network mentioned in Section 5.4, additional graph theory can be applied to enrich the data. In graph theory, it is said that if node **A** is a neighbor of node **B**, and node **A** is also a neighbor of node **C**, it is likely that, at some point in time, nodes **B** and **C** will be connected, too. This phenomenon is referred to as *triadic closure*, and is proven to hold for most types of (social) networks [7].

In Figure 13, triadic closure is applied to the simple network drawn in Figure 11. Using this theory, an educated guess can be made as to which users will be connected at some point in time (but are not connected yet). Taking this out of the context of public documents on the domain, this information can be used to establish a list of users which likely know each other already, or are socially close (but are not specifically listed as such in the metadata of any file, yet). This list can then be used for phishing campaigns against the organization in question (e.g. by means of sending fake LinkedIn invitations amongst those pairs of users).

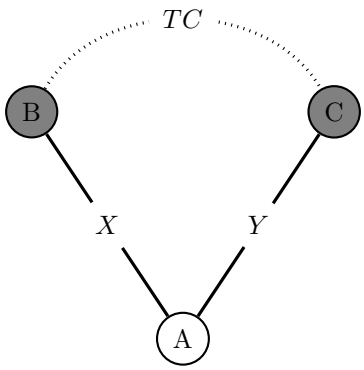

*Figure 13: If user **A** "knows" user **B**, and user **A** "knows" user **C**, it is likely that, at some point in time, users **B** and **C** will connect (denoted as "TC" in this graph). This is inferred by triadic closure[7] and can be used for crafting phishing attacks.*

5.4.3 Clustering coefficient

The extent to which triadic closure has already taken place in a graph can be measured, and can tell something about the probability that triadic closure will happen for a given set of nodes (which are potential subjects to triadic closure) in that graph in the future [8] [20] [21].

To do this, the *clustering coefficient* can be calculated for each node in a network. This was done for the edit network as well, and added to the information per domain. This resulted in a pair of users subject to triadic closure, and their average clustering coefficient (i.e. the likelihood they will indeed connect). To get to this value, for each user, all unique pairs of neighbors were listed. Then, for each pair of neighbors, it was reviewed whether they were already connected or not. If a node has six pairs of neighbors, for instance, and only two are connected, the resulting clustering coefficient for that node is 2/6, or 1/3.

In Figure 14, an example of a node with an extremely low clustering coefficient is given, whereas in 15, the node's clustering coefficient is 1.0 (the highest value possible). It is assumed, then, that users which ideally both have 1.0 as their clustering coefficients, are highly likely to connect. Users who both have 0.0 as their clustering coefficients, on the other extreme end, are not likely to connect. Anything in between is to be interpreted by the people doing the analysis case-to-case.

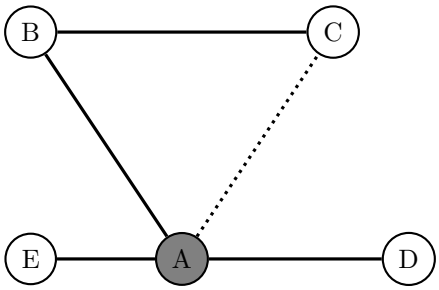

*Figure 14: In this example, nodes (users) **A** and **C** are subject to triadic closure. None of its neighbors (**{B,D,E}**) are connected, so the clustering coefficient of node **A** is 0.0. It is thus not likely that **A** and **C** will connect (when taking this measure into account).*

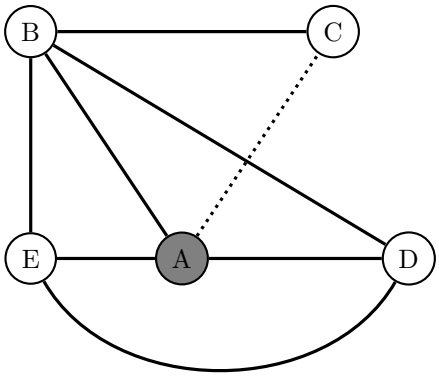

Figure 15: In this example, nodes (users) ***A*** *and* ***C*** *are subject to triadic closure. All of its neighbors (****{B,D,E}****) are connected, so the* ***clustering coefficient*** *of node* ***A*** *is 1.0. It is thus* ***highly*** *likely that A and C will connect (when taking this measure into account).*

It is also good to note that leaf nodes (i.e. nodes with only one neighbor) are left out of this calculation as for those, no pair of neighbors could be listed.

After applying the clustering coefficient for each node to all pairs of suspects to triadic closure (as described in Section 5.4.2), it turned out there were 210 unique pairs of users with an *average* clustering coefficient of 1.0 (which means they were both at, or close to the maximum value). These users are thus most likely to connect, and could be used as phishing targets. A quick review of the list revealed that, indeed, some of those users were likely to be socially very close in real life. For example, the users in one of these pairs were both in the board of a committee where they were in meetings together (which was found through a Google search). No further validation of the list was done due to time constraints, however. Therefore, it is hard to conclude something based on this with a high certainty.

In Figure 16, the distribution of clustering coefficient for all pairs of users subject to triadic closure is shown. Most of the pairs have an average clustering coefficient between 0.0 and 0.1, which indicates this subset of the edit network is relatively *sparse*.

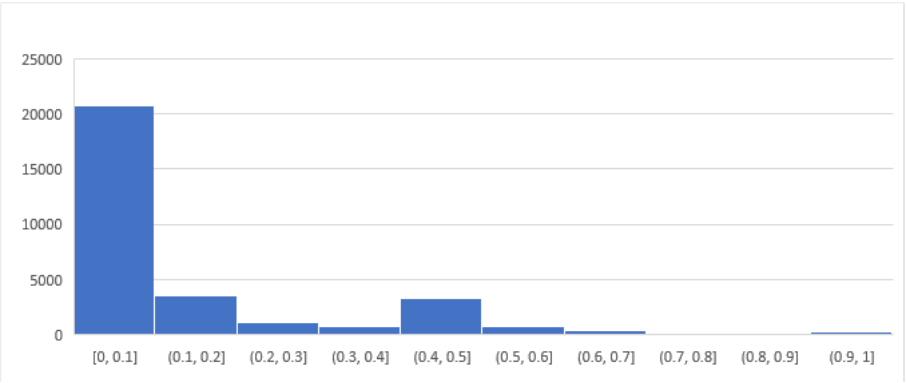

Figure 16: The distribution of the average clustering coefficient for the pairs of users from the list established in Section 5.4.2. In total, the list contained 30,987 unique pairs of users, with an average clustering coefficient ($0.0 \geq X \leq 1.0$*). By far, most users have an average clustering efficient of* $0.0 \geq X < 0.1$*, which indicates this subset of the edit network has a relatively low density.*

While the triadic closure offers new insights into *implicit* relationships amongst users in the edit network, the *explicit* relationships between users (i.e. two users who worked together on the same document) can also be used for phishing as-is. As it turns out, users are around 4.5 times more likely to click on phishing links sent by existing contacts [10][12]. The information on explicit relationships shows just that.

5.4.4 Interconnected domains

Edit networks can be created for each domain, but also for the entire dataset as a whole. When doing that, it appears there exist domains which share one or more nodes. For example, one domain had a user who had created a document, which was later edited by someone on another domain. In short, this user functioned as a *bridge* between the two domains. This can be interesting for hackers who want to traverse from one domain to another. As the 2017 NotPetya attack showed, lateral movement after a successful breach can quickly have extremely destructive effects [16], so it is not unlikely that attackers will look for ways to do so in the future, too. Therefore, finding these nodes seems relevant.

It is important to note that there can be false positives in this case, as really common usernames (such as the user "Default User" or "Province of X"), or users who share a username by coincidence, erroneously show similarities between domains. In practice, however, it turned out to be relatively easy to spot those false positives while reviewing the visualized edit networks manually.

5.5 Email addresses

Using a regular expression, 11,057 email addresses were found in the dataset. This includes email addresses that were in some place present in a metadata tag value. After parsing those email addresses, it turned out that they were from 893 different domains.

Interesting here is that many of these email addresses were from domains that were not in the initial scope. For example, 8 unique *@hotmail.com* addresses were found, which *could* indicate people working with shadow systems in IT. The controversy around Hillary Clinton[17] showed this is not unprecedented in governments.

Something else email addresses can reveal information about is external contacts of a specific organization. If an email address belonging to a contractor is still present in the metadata of a document, it can mean that this specific contractor worked for the organization the document belongs to. This was not reviewed in this research, however. More information about external companies can be found in in Section 5.9.

[17] https://en.wikipedia.org/wiki/Hillary_Clinton_email_controversy

5.6 Mapping out file systems

After analyzing the metadata of the hundreds of thousands of files, it turned out there were many references to file paths. These file paths often started with *C:*, indicating it was written on a Windows file system. The file paths were listed in a variety of tags: in many cases even, the title consisted solely of the original file path.

This information was combined per domain, and visualized to get more insight into the internal file systems of the organization in question. In Figure 17, the Windows file paths were aggregated and visualized for a domain of the dataset. In the dashboard (see Section 5.11), this visualization is automatically present for each analyzed domain.

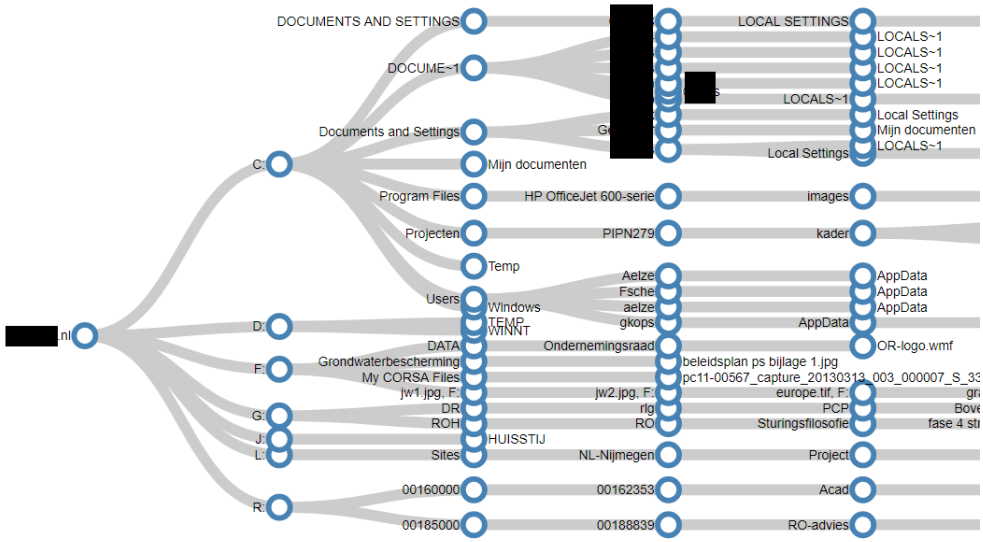

Figure 17: A real-life example of Windows file paths mapped out for a domain from the dataset. While the "C:" and "D:" drives are probably local drives, partitions starting with "L:" and "R:", in this example, likely refer to mounted network shares.

As can be seen, there are regular "C:" and "D:" drives, as well as more uncommon drives, such as "L:" and "R:". This strongly points towards network shares of some kind, which generally get assigned a letter relatively high up in the alphabet.

In total, 3,071 URLs starting with an uppercase or lowercase letter (*[a-zA-Z]*), followed by *:*), were found. These are all Windows file paths. The full list of drive names can be found in Appendix C.

Additionally, Samba shares (identified by a URL starting with "smb:") were found in three cases. This can be extremely interesting for a hacker for the same reasons mentioned in Section 5.4.4, that is, infection spreading after obtaining an initial foothold.

In 40 cases, a *.local* URL was encountered. As defined in RFC 6762, this pseudo-top-level-domain is only used on local area networks [5]. This means those URLs are also, by definition, not routable from outside the organization's network. Therefore, they should not have been there and can safely be assumed to be cases of unintended information leakage. For an attacker, this information is valuable as it gives insight into the intranet of the organization.

5.7 Matching CVEs and CPEs

In the National Vulnerability Database (NVD)[18], the National Institute of Standards and Technology (NIST) keeps track of all reported vulnerabilities. These vulnerabilities are referred to as Common Vulnerabilities and Exposures (CVE), and the software and hardware platforms they affect are called Common Platform Enumeration (CPE). As the *Creator Tool* metadata tag shows information on the software a document was created with, it was reviewed whether it was possible to match those software versions with a known CPE. In Appendix B, the top 50 of most common values of the "Creator Tool" tag is listed.

After downloading the public CVE and CPE feeds from the NVD, a script was written to check how "similar" a software version from the metadata (the "Creator Tool" tag) was to all known CPEs (using *fuzzywuzzy*[19]). Then, the CPE with a title most similar to that (with a similarity ratio of above 96 percent[20]) was said to be a matching CPE for that software version.

Using this approach, potential CVEs were found for almost all domains, as it turned out the *Microsoft Office* software was relatively easy to match and this software is commonly used. However, it is much likely that there are a lot of false positives here. This is due to two aspects: **time** and **versioning**.

First of all, in this approach, time was not taken into account. This means that an old software version may have been used to create a document back in 2013, and that a CVE for this software version was published by NIST in 2017. However, it can be that, in the meantime, the software is not used anymore. The CVE will then still show up.

Secondly, the metadata of most files does generally not give a detailed description of the *exact* versions used. Often, subversions are left out, which makes it impossible to see what patches have or have not been applied by a specific organization. This is a clear limitation on what can be retrieved from metadata.

In Figure 18, a real-life example is shown for one of the domains. In the example, most of the matched CVEs are matched because the domain hosted public documents created using *Microsoft Word 2016*. This also shows the versioning limitation mentioned earlier; it is impossible to know what patches the organization applied.

In total, 98 CPEs were found to match a specific software version found in metadata of files. This list of 98 CPEs was verified manually, and checked for validity. It turned out that **79** were indeed correct matches, whereas **19** were false positives, where, for instance, one digit difference in the version number (e.g. 1.13 vs. 1.12) resulted in a high similarity ratio ($> 96\%$). This is to be taken into account when using this approach.

In Table 4, a summary of the number of CPEs, "Creator Tools" and the validity of the results is given. There are two numbers for the number of unique creator tools: the first one is the raw data, the second one excludes all invalid creator tools (such as ones that had the title of the document listed as the creator tool, which cause a relatively long tail in the distribution). All in all, it can be said that roughly ten percent of the unique creator tools could successfully be matched to a known CPE.

Using these CPEs, in 5,952 cases, a CVE was assigned to a domain, of which 69 were unique. This match between CPEs and CVEs is considered to be valid, as NIST published them as such. Take into account, however, that this includes the previously mentioned falsely matched CPEs, which were 19.

Lastly, it is good to know that, in this approach, domains can be assigned the same CVE more than once. This happens if a CVE affects multiple CPEs, which the domain has all linked to it.

[18] https://nvd.nist.gov/

[19] https://pypi.python.org/pypi/fuzzywuzzy

[20] this number was chosen after trial-and-error; it turned out not to be feasible to measure the exact validity of the script automatically and scalably

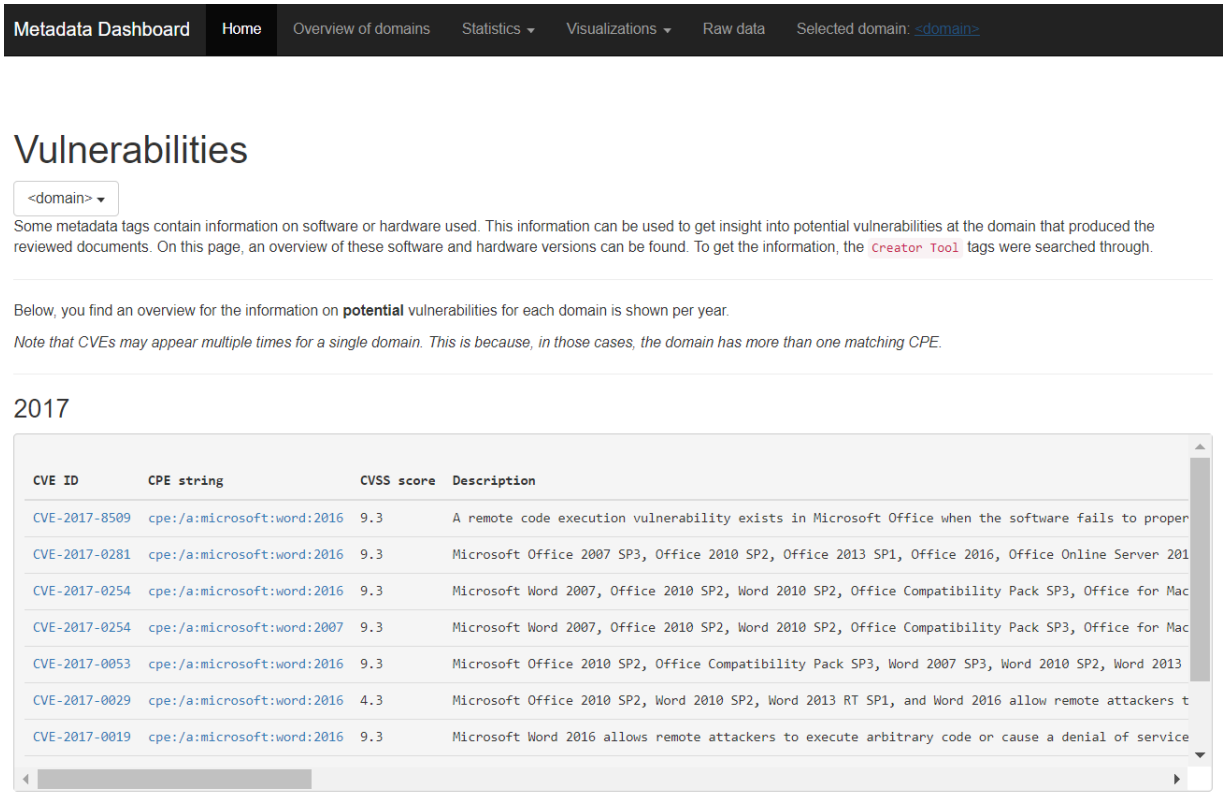

Figure 18: A real-life example of the potential vulnerabilities found for one of the domains from the dataset. This also underlines the versioning problem mentioned in section 5.7; it cannot be verified what patches have or have not been applied by the organization by just looking at the metadata of the documents.

Number of unique CPEs	118,371
Number of unique "creator tools"	6,340
Number of unique "creator tools" (after quick sanitization)	785
Correctly matched to CPE	79
Falsely matched to CPE	19
Total number of CVEs	5952
Number of unique CVEs	69

*Table 4: The results of attempting to match software versions from the **Creator Tool** metadata tag directly to CPE titles as published by NIST (using a similarity ratio threshold of 96). The third row in the table excludes invalid "Creator Tools", such as ones that had the document's title listed as the Creator Tool. Using **all** matched CPEs, 69 unique CVEs were found.*

5.8 Underlying (operating) systems

Searches for "Windows Server" showed the *Producer* tag in some cases contained the name of PDF creation software, along with the Windows Server version it was running on (or at least designed for). The three main programs that showed this behavior were *PDF-XChange Printer*, *novaPDF Lite Server* and *pdfFactory*.

Below, an example of the most commonly found string of this type can be seen:

> *PDF-XChange Printer 2012 (5.5 build 315) [Windows Server 2008 R2 x64 Server 4.0, Enterprise Edition (Build 7601)*

This particular example was found 852 times, and shows the exact subversion, which is interesting as it is normally not the case in metadata (see Section 5.7 for more information).

Using this information, one can have a look at the last time a particular version string was seen and decide whether it can be counted as a vulnerability or not. For example, it turned out Windows Server 2003 R2 was still used by one domain in May 2017. In Figure 19, it is shown how many documents had "Windows Server 2003" in their metadata, and when they were created (within the period 2007-2017). This shows files were still created after the official end-of-life (14 July 2015 [21]). This is considered a vulnerability, as Microsoft no longer supported it after that date. In Figure 20, the same graph is plotted for "Windows Server 2008", but without an end-of-life indicator as it is still supported[22].

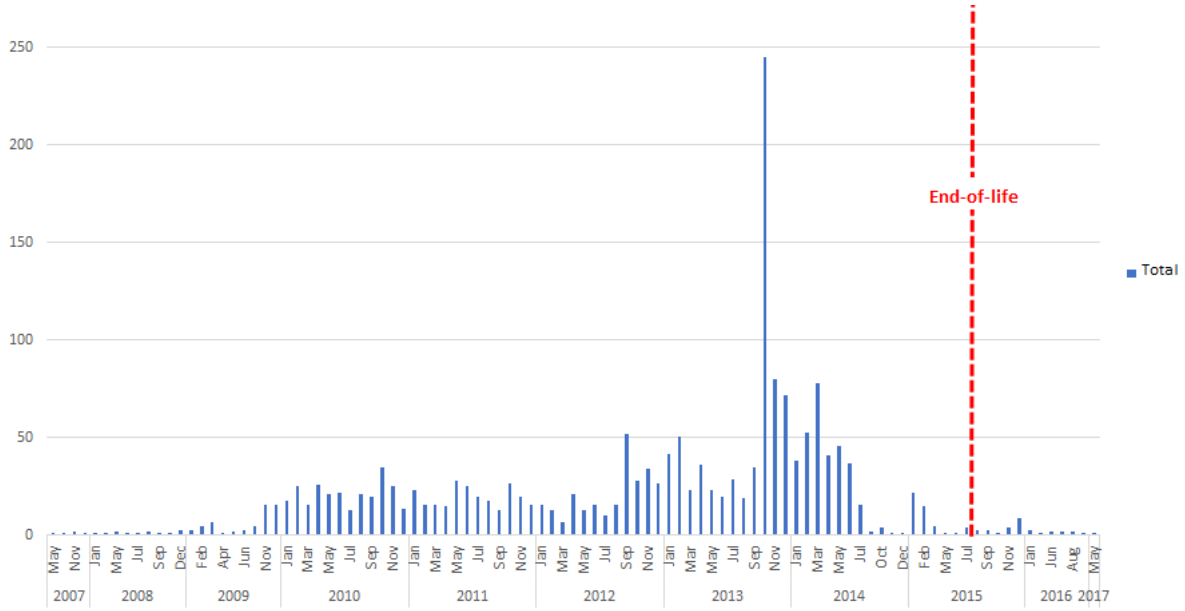

Figure 19: The number of files with "Windows Server 2003" in their metadata, along with their creation dates. Everything after 14 July 2015 is considered vulnerable. Note: only the period 2007-2017 is shown here.

[21] https://www.microsoft.com/nl-nl/cloud-platform/windows-server-2003
[22] https://support.microsoft.com/en-us/lifecycle/search/?p1=14134&wa=wsignin1.0

5.9 (External) Companies

In 39,836 cases, the "Company" metadata tag was populated. As its name suggests, this tag holds information on what company created a document. Interestingly, this does in many cases not match the organization owning the domain. What this means is that another company wrote a document, which was then published on the domain the document was downloaded from.

In total, 2,204 unique companies were found this way. A quick look through the data shows that some third parties, which, for instance, are big consultancy firms, are associated with hundreds of files. Attackers might find this information useful, as it says something about the parties the organization in question deals with regularly and could thus be an interesting attack vector.

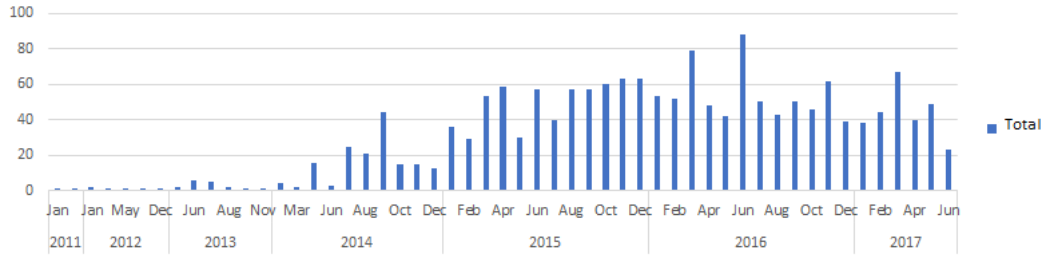

Figure 20: The number of files with "Windows Server 2008" in their metadata, along with their creation dates.

5.10 Further enriching the user data

As mentioned in Section 5.4, the *Creator*, *Author* and *Last Modified By* metadata tags contain the usernames of the users who worked with specific documents. Analysis showed the usernames often contain (parts of) first and last names of users. When applying the techniques and theories described in Sections 5.4.1, 5.4.2 and 5.4.3, it can prove to be useful to have more information on these people. In this section, methods for doing this are proposed.

5.10.1 theHarvester

For each domain, a tool called *theHarvester*[23] was run to search for more information on a specific domain online. This was all done passively and automatically. One of the things *theHarvester* outputs is a list of email addresses found for a domain. By manually looking at what those email addresses look like (i.e. how they are constructed), one can get insight into what the most likely email format used for the domain is. The next step is to apply this logic to users who are to be targeted for phishing.

One thing to mention is that *theHarvester* did not always present extensive results. For some domains, only *info@* or other general email addresses are returned. For others, there are tens of email adresses. In either case, it proved to be a useful add-on for getting a broader picture of the domain reviewed.

5.10.2 Anti Public Combo List

Another good source of information is the *Anti Public Combo List*[24]. This list contains over 450 million email addresses (and 560+ million records). At the moment of the analysis, a search for all domains in the dataset was also done here. The passwords in the data breach were not included for ethical reasons.

Using a custom script, the search results in the Anti Public Combo List were reviewed for each domain, and the email syntax was parsed from that. In Figure 21, a real-life example of the parsed email syntaxes for a domain from the dataset is shown. With this list, and a random user called *John Doe*, it seems safe to guess that the most likely email address belonging to that user is something like "jdoe@¡domain¿".

[23] https://tools.kali.org/information-gathering/theharvester
[24] https://haveibeenpwned.com/PwnedWebsites#AntiPublic

Email syntax	Number of occurrences
{f}{last}	100
{f}{prefix}{last}	29
{first}{prefix}{last}	13

Figure 21: The real-life example of the output of the script parsing the search results from the "Anti Public Combo List" for a specific domain.

Slightly outside the ethical limitations of this project, it is good to note that attackers *would* probably take the passwords in account. Password reuse is considered common behavior [11] [6], which shows this is a relevant threat to the domains in this dataset, too.

5.10.3 Adding dorks

As a last step, dorks are added for each domain in the dashboard (see Section 5.11). This enables one to quickly do an effective Google search for a specific user, CVE, creator tool, et cetera. As an example, Google can be used to search for a specific user on LinkedIn, in conjunction with the domain he was linked to[25].

5.11 Metadata Dashboard

As the number of domains and files in the dataset was too large to analyze manually, there was a clear need for data aggregation. Therefore, a "Metadata Dashboard" was built. The scripts running the scans and the metadata analysis produce their output such that the dashboard can directly use and visualize the data. The code for the dashboard was made fully open source, and is available at https://github.com/AukeZwaan/metadata-dashboard.

[25] https://www.google.nl/search?q=site:linkedin.com+"JohnDoe"+"Microsoft"

6 Discussion and conclusion

It is clear that metadata extracted from public documents can be used in the reconnaissance phase of a cyber attack. It is also clear that information leakage through these metadata are often unintended. The large number of usernames shows a lot about the organization creating the documents.

While the tools published with this research, along with the dashboard, are not "attack tools" on their own, they do give a relatively broad insight into an organization of which the domains are reviewed.

Not only does the metadata reveal what user made a specific document, it also shows inter-relationships amongst those who created files, and those who edited them. By aggregating and visualizing the users who created documents, and their interrelations (i.e. what users worked on a document together), one has the ability to create an "edit network", which gives insight into social connections inside the organization. This information can then further be enriched by applying graph theory. Eventually, an educated guess can be made as to what users are likely to be "good" victims for spearphishing attacks. For this, spatial information on the surroundings of specific nodes in the graph is required.

In general, this information about users is ubiquitous, about 65 percent of all documents having a populated "Author" tag, and over 85 percent having a populated "Creator" tag. On critical remark here is that the underlying assumption of the triadic closure theory is that creators and modifiers are acquaintances. This means that the theories proposed in this paper only apply in situations where no large numbers of "modifiers" were in between.

Further network analysis showed that the edit network was relatively *sparse*, meaning the density was low. This is because the *Last Modified By* tag was not often present, and many people just edited one single document with each other. Still, *bridges* between different domains were found. These nodes, or users, could be used by attackers to traverse to another domain.

While the findings from this research may seem to only be a problem of the organization hosting the domains, activity (i.e. creations and modifications of documents) can be plotted per user, too. This is shown to reveal private information on users working for the organizations. For example, it was shown that working days and behavior during those days can be measured and compared with others.

Additionally, it was shown that, based on the "Creator Tool" metadata tags, something can be said about whether or not an organization might potentially be subject to existing vulnerabilities as published by NIST. While this is a valuable source of information, it must be said that there are a lot of false positives due to the limited ability to properly match "Creator Tools" and CPEs. This is caused by the fact that metadata does contain software versions, but no detailed *subversions*. This makes it hard to get valid results on this point (and to really pinpoint the best attack vector). However, it might still be preferable over having to guess blindly for an attacker.

Another type of metadata found throughout many domains was information on file systems. On many domains, file paths could be found, which not only showed that Windows was used, for instance, but also what partitions and (sub)directories were present at the time documents were created. It was shown to be possible to map out file systems like this, and to get a clear overview on where the documents resided before they were published. In some cases, even, Samba shares were found. Other revealing file paths included URLs to local intranets of organizations. This information can prove to be useful after an attacker gets an initial foothold in the organization, and wants to do lateral movement without first having to scan the entire network, for example.

Finally, it turned out to be possible to use third party information to predict email addresses of users found in metadata. This was done by searching for email addresses in data breaches, and deriving the most common email syntax for each domain from the results. Using this technique, the usernames found in the metadata can be converted into valid email addresses, which can, in turn, be used for (spear)phishing.

All in all, it can be concluded that the Dutch government leaks metadata at almost all domains. On average, hundreds of files were hosted on websites, and there were no clear signs of metadata being stripped out preemptively. In almost all documents, at least temporal information on creation and modification could be found, as well as a user who created or edited it.

7 Recommendations

Considering the fact that metadata leakage was proven to be so ubiquitous for government domains, it is recommended that action is taken. In the current situation, unnecessary information on users is shared with anyone capable of downloading any document from a government domain.

Attackers could do wrong with this information. Not only does it give them information on who worked at specific organizations at what time, it also provides an easy way to craft a really powerful (spear)phishing campaign against the organizations behind the domains. It shows new attack vectors (i.e. companies that were shown to have a connection with the reviewed domains), and using specific analysis techniques, potential vulnerabilities in the underlying systems can be pinpointed with relative ease. While no *exact* software versions can be derived from the metadata directly, it *does* give attackers insight into the types of software used, and what main versions are running.

As shown in Section 5.3, private information of employees can be gathered from the public documents reviewed in this research. It is therefore recommended to review whether keeping this metadata stored in public metadata is in accordance with current local legislation of the organizations in question. With the upcomping GDPR and the "right to be forgotten", which is already in place in the European Union, this can turn out to become a serious problem if not dealt with properly [14].

The simplest approach in addressing the issues would be to simply "strip out" all unnecessary information from the published documents which are already online. The next step is to filter out any metadata of *new* documents which are yet to be published. Considering the information leakage one is proven to face otherwise, it is definitely recommended to take these steps to reduce any risks.

8 Future work

8.1 Analyzing file contents

While the metadata was only looked at here, it is probable that the file contents themselves also contain valuable information. This step can easily be introduced between downloading the file and overwriting it with the output from *exiftool*. One tool that can be used to get more out of documents is *bulk_extractor*[26], which searches for known patterns using regular expressions. Custom regular expressions can be added, too. One thing to take into account here, is that it is best to first try to get plaintext from a given file, which software suites like *PDF Tools*[27] can do, for example.

8.2 Applying passive scanning

Right now, it turned out not to be feasible to apply passive scanning. However, in the released set of tools, a script to perform "Bing Dorks" is included. It introduced random delays (between 5 and 15 seconds) and picks a user agent from a list of valid user agents each time it does a request. However, the tool still resulted in an IP block.

If it can be made working again, one way to go is to store the output of both the active and passive scans, after which they can be merged and all unique URLs are kept.

Another point to add here is that passive scanning could add even more value if it is used for the links the active scanner cannot find by definition (see Section 4.2.2). These backlinks could previously be found using Google's *link* operator, but a Google employee recently told webmasters not to use it anymore[28].

8.2.1 theHarvester

In the current approach, an initial scope was defined and was kept that way. However, tools like *theHarvester*, which was also described in Section 5.10.1, can be used to see what other websites reside on a specific server, for example (i.e. shared hosting). Theoretically, a successful breach on one of those websites can mean an attacker can pivot into the website he was initially looking for, too.

Using websites like Robtex.com[29], the public infrastructure of the websites can be looked up. However, this is not very scalable. A scope over a thousand domains can thus be a hassle.

8.3 Extending the social graphs

The dashboard currently only shows the edit network, and creates a list of potential phishing targets using the data in the graph.

To extend this, one could try to look up more information about the targets. To do so, first, one would look up the pairs of users with the highest probability of matching. Then, the real-life friends are searched for using social media. After gathering the data for many users, it can be reviewed who is connected to whom. Eventually, the triadic closure principle here, too, and even verify the triadic closure proposed by the network analysis (see Section 5.4.2).

8.4 Additional file types

In the current scope of research, the list of "interesting" file types is relatively small. It can be reviewed whether additional file types can offer more valuable information. For example, *.docm* files can indicate an organization is used to work with macro-enabled documents, which is an important attack vector for modern malware. It is worth mentioning here that Google cannot be used to passively search for those, as *.docx* files are returned for the *filetype:docm* operator, too. However, the active scanner *can* of course simply look for the *.docm* extension. A quick search through the current dataset shows that some original titles of documents include that extension, but were saved as a regular *docx* file later (therewith stripping out any macros). This is an indicator that there might be more beneath the surface.

[26] www.forensicswiki.org/wiki/Bulk_extractor
[27] https://blog.didierstevens.com/programs/pdf-tools/
[28] https://twitter.com/JohnMu/status/819094559770738688
[29] https://www.robtex.com/dns-lookup/

8.5 Adding temporal information

Temporal information can make the results more accurate. As mentioned in Section 5.7, CVEs and CPEs are currently looked for in the *entire* dataset. This means that some of the tools which triggered the detection of a specific CVE might no longer be actively used by the organization in question. Therefore, it would be good to show what software was in use at what specific time. Using these temporal features, one can more accurately find out what still applies today.

8.6 Company analysis

As mentioned in Section 5.9, information on companies creating or modifying documents on behalf of the organizations behind the domains can be found in metadata, as well. By looking at which companies work for an organization from time to time, one might even go as far as deriving new partnerships from this. This information might be really valuable for shareholders of the companies involved.

Additionally, a mapping between companies and the domains (or organizations) they work for can be made. This information can also be derived from other sources, such as (shared) hosting providers and web development companies.

References

[1] Google bot going mad on my site, 10 pages/sec. https://www.webmasterworld.com/google/4257070.htm. Accessed: 2017-06-07.

[2] Google indexes stack overflow at a rate of 10 requests per second — hacker news. https://news.ycombinator.com/item?id=2382728. Accessed: 2017-06-07.

[3] How can i tell google that i'm ok with more than 10 requests per second to my website? https://productforums.google.com/forum/#!msg/webmasters/rlFZNBi4gYA/2JD4QUjwwYsJ. Accessed: 2017-06-07.

[4] C. Alonso, E. Rando, F. Oca, and A. Guzmán. Disclosing private information from metadata, hidden info and lost data, Aug 2008.

[5] S. Cheshire and M. Krochmal. Multicast dns. RFC 6762, RFC Editor, February 2013. http://www.rfc-editor.org/rfc/rfc6762.txt.

[6] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The tangled web of password reuse. In *NDSS*, volume 14, pages 23–26, 2014.

[7] D. Easley and J. Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[8] P. W. Holland and S. Leinhardt. Transitivity in structural models of small groups. *Comparative Group Studies*, 2(2):107–124, 1971.

[9] M. Honan. Oops! did vice just give away john mcafee's location with photo metadata?, Dec 2012.

[10] J. Hong. The state of phishing attacks. *Communications of the ACM*, 55(1):74–81, 2012.

[11] B. Ives, K. R. Walsh, and H. Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4):75–78, 2004.

[12] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[13] V. Joler and A. Petrovski. Metadata investigation : Inside hacking team, Oct 2015.

[14] A. Mantelero. The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review*, 29(3):229–235, 2013.

[15] J. J. Migletz. Automated metadata extraction, 2008.

[16] msft-mmpc. New ransomware, old techniques: Petya adds worm capabilities. Accessed: 2017-07-05.

[17] Open State Foundation. Veilig http (https). https://pulse.openstate.eu/https/guidance/. Accessed: 2017-07-01.

[18] L. Pesce. Document metadata, the silent killer..., March 2008.

[19] Vice Staff. We are with john mcafee right now, suckers. https://www.vice.com/en_us/article/we-are-with-john-mcafee-right-now-suckers. Accessed: 2017-06-12.

[20] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

[21] H. Yin, A. R. Benson, and J. Leskovec. Higher-order clustering in networks. *arXiv preprint arXiv:1704.03913*, 2017.

Appendices

A Top 50 of domains with most files downloaded and processed

Domain	Number of files
rijksoverheid.nl	49548
examenblad.nl	47354
brabant.nl	21367
cbs.nl	18742
overijssel.nl	16310
steenwijkerland.nl	13228
limburg.nl	10158
afm.nl	10014
pvda.nl	8480
naktuinbouw.nl	7988
geldermalsen.nl	7373
d66.nl	7018
amsterdam.nl	5346
tuinbouw.nl	5087
alblasserdam.nl	4984
nationaleombudsman.nl	4892
gemeente-steenbergen.nl	4833
achtkarspelen.nl	4714
beesel.nl	4412
aalten.nl	4135
nvwa.nl	4005
hengelo.nl	3841
fryslan.frl	3612
platformparticipatie.nl	3594
onderwijsraad.nl	3495
noord-holland.nl	3386
nvao.net	3326
schielandendekrimpenerwaard.nl	3137
onderwijsinspectie.nl	3043
hdsr.nl	2967
igz.nl	2961
officiele-overheidspublicaties.nl	2710
amstelveen.nl	2643
helmond.nl	2570
helpdeskwater.nl	2398
neerijnen.nl	2393
defensie.nl	2271
hendrik-ido-ambacht.nl	2180
cvdm.nl	2085
rotterdam.nl	2038
rekenkamer.nl	1941
ilent.nl	1890
zuid-holland.nl	1872
appingedam.nl	1710
rijkswaterstaat.nl	1647
cbho.nl	1604
zorginstituutnederland.nl	1587
renkum.nl	1585
pianoo.nl	1541
poraad.nl	1523

Table 5: The top 50 domain with most files downloaded and processed

B Top 50 of "Creator Tools"

Creator Tool	Occurrences
PScript5.dll Version 5.2.2	21876
Acrobat PDFMaker 7.0 voor Word	3868
Acrobat PDFMaker 10.0 voor Word	3335
Microsoft® Word 2013	2048
Microsoft® Word 2010	1785
Acrobat PDFMaker 9.1 voor Word	1562
Neevia Document Converter 5.1.0	1184
Microsoft® Office Word 2007	1101
Acrobat PDFMaker 10.1 voor Word	1094
Adobe InDesign CC 2015 (Macintosh)	1008
PScript5.dll Version 5.2	927
Adobe InDesign CS6 (Macintosh)	877
Acrobat PDFMaker 9.0 voor Word	631
Acrobat PDFMaker 11 voor Word	615
Adobe InDesign CS4 (6.0)	591
Adobe InDesign CS5.5 (7.5.3)	520
Adobe InDesign CC 2017 (Macintosh)	520
Adobe InDesign CS4 (6.0.6)	431
Adobe InDesign CS6 (Windows)	423
Adobe InDesign CS2 (4.0.5)	421
Image2PDF Command Line Software	420
PDFCreator Version 1.7.1	402
Adobe InDesign CS3 (5.0.4)	349
Adobe InDesign CC 2014 (Macintosh)	330
Acrobat PDFMaker 6.0 voor Word	275
Adobe InDesign CC 2015 (Windows)	274
Microsoft® PowerPoint® 2010	271
ABBYY FineReader Engine 10	251
Adobe InDesign CS5 (7.0.3)	247
Canon iR-ADV C7280 PDF	244
PDFCreator 2.3.0.103	213
Adobe InDesign CS5 (7.0)	207
Acrobat PDFMaker 8.1 voor Word	205
Microsoft Word	193
Adobe InDesign CS2 (4.0.4)	188
UnknownApplication	178
Adobe InDesign CS5.5 (7.5)	178
Kofax Ascent Capture	175
PDFCreator Version 1.2.0	174
XPP	172
Canon	172
Adobe InDesign CS5.5 (7.5.1)	170
PDFCreator Version 1.3.2	159
MicroStation 8.11.7.443 by Bentley Systems, Incorporated	139
Adobe InDesign CS5.5 (7.5.2)	137
Canon iR-ADV C5240 PDF	134
Adobe Acrobat Pro 9.0.0	134
PDFCreator Version 0.9.3	132
PDFCreator Version 1.7.2	127
Xerox WorkCentre 7556	125

Table 6: The top 50 most commonly used "Creator Tools" (i.e. the values of the "Creator Tool" metadata tag).

C All drive names (partition letters)

Partition letter	Occurrences
C	1,031
J	802
P	448
H	234
D	233
Y	208
X	143
G	137
F	122
M	94
K	65
d	64
N	48
L	48
I	43
Z	40
z	37
E	32
O	27
T	21
V	19
R	19
Q	12
W	11
c	11
S	9
m	5
f	3
A	3
p	2
g	1

Table 7: The names of file drives, and how often they were found within the dataset. There are relatively many "uncommon" drives, which could very well indicate file shares.

D Domains with files analyzed

- 2todrive.nl
- 50pluspartij.nl
- a13a16rotterdam.nl
- aaenhunze.nl
- aaenmaas.nl
- aalburg.nl
- aalsmeer.nl
- aalten.nl
- aanbiedersmedicijnen.nl
- aanpakjeugdwerkloosheid.nl
- aanpakregeldruk.nl
- aanvalopoverval.nl
- achtkarspelen.nl
- acm.nl
- acwet.nl
- adviescommissiewater.nl
- aerius.nl
- afm.nl
- agendastad.nl
- agentschapszw.nl
- agentschaptelecom.nl
- agressievrijwerk.nl
- agroberichtenbuitenland.nl
- agv.nl
- ahn.nl
- aivd.nl
- alblasserdam.nl
- alertonline.nl
- algemenebestuursdienst.nl
- alkmaar.nl
- alleenjijbepaalt.nl
- allesisgezondheid.nl
- almelo.nl
- almere.nl
- alphenaandenrijn.nl
- alphen-chaam.nl
- ameland.nl
- amersfoort.nl
- amstelveen.nl
- amsterdam.nl
- antennebureau.nl
- apeldoorn.nl
- appingedam.nl
- arboportaal.nl
- architectenregister.nl
- arnhem.nl
- assen.nl
- atlasleefomgeving.nl
- atlasnatuurlijkkapitaal.nl
- atnfi.nl
- autoriteitnvs.nl
- autoriteitpersoonsgegevens.nl
- awti.nl
- baarle-nassau.nl
- basisregistratiesienm.nl
- bedrijvenbeleidinbeeld.nl
- bedum.nl
- beeldmateriaal.nl
- beemster.net
- beesel.nl
- belandadananda.nl
- belastingdienst-cn.nl
- belastingdienst.nl
- beleefdedeltaroute.nl
- bergen-nh.nl
- beschikbaarheidbijdrage-medische-vervolgopleidingen.nl
- bestrijdingsmiddelen-omwonenden.nl
- bewegingdenk.nl
- bics.nl
- bigregister.nl
- biobasedeconomy.nl
- biociden.nl
- biodiversiteit.nl
- bkd.eu
- bkwi.nl
- bodemrichtlijn.nl
- brabant.nl
- brabantsedelta.nl
- brunssum.nl
- bunnik.nl
- bureaugateway.nl
- bureauicttoetsing.nl
- c2000.nl
- cannabisbureau.nl
- cbg-meb.nl
- cbho.nl
- cbi.eu
- cbs.nl
- ccmo.nl
- cda.nl
- ceaweb.nl
- centralecommissiedierproeven.nl
- cfto.nl
- challengestad.nl
- checkjeschoolgebouw.nl
- checklistschoonmaak.nl
- chemischestoffengoedgeregeld.nl
- christenunie.nl
- cibg.nl
- civ-voetbal.com
- cloudfront.net
- clu-in.org
- coa.nl
- codexalimentarius.nl
- coevorden.nl
- cokz.nl
- collegesanering.nl
- collegevanrijksadviseurs.nl
- commissiegeweldjeugdzorg.nl
- commissievanaanbestedingsexperts.nl
- communicatierijk.nl
- compensatieregelingpgb.nl
- consuwijzer.nl
- contactpuntbouwproducten.nl
- cpb.nl
- criminaliteitinbeeld.nl
- ctgb.nl
- ctivd.nl
- cultureelerfgoed.nl
- cultuur.nl
- cultuursubsidie.nl
- cvdm.nl
- cvta.nl
- cvte.nl
- cybersecurityraad.nl
- d66.nl
- daargeefjeom.nl
- dalfsen.nl
- dcypher.nl
- debilt.nl
- defensie.nl
- defensiepijpleidingorganisatie.nl
- dekinderombudsman.nl
- deleerkaart.nl
- delerarenagenda.nl
- deltacommissaris.nl
- demarne.nl
- denationalepensioendialoog.nl
- denhaag.nl
- denieuwepraktijk.nl
- derijkscampus.nl
- dezorgagenda.nl
- dggf.nl
- dictu.nl
- dienstterugkeerenvertrek.nl
- diergeneeskunderegister.nl
- digicommissaris.nl
- digid.nl
- digitaleoverheid.nl
- digitoegankelijk.nl
- dji.nl
- dnb.nl
- doc-direkt.nl
- doetinchem-wesel380kv.nl
- dommel.nl
- donorgegevens.nl
- donorregister.nl
- dordrecht.nl
- drenthe.nl
- dsta.nl
- duo.nl
- duo-onderwijsonderzoek.nl
- dus-i.nl
- dutchdigitaldelta.nl
- dutchh2.nl
- dutchhorticulture.nl
- duurzaamdoor.nl
- duurzaamgww.nl
- dwangindezorg.nl
- earonline.nl
- eceonline.nl
- edagcbg.nl

- eerenvrijheid.nl
- effectiefarmoedebeleid.nl
- eherkenning.nl
- eijkelkamp.com
- energieplein20.nl
- eranetbioenergy.net
- erfgoedinspectie.nl
- erfgoedmodernetijd.nl
- ertms-nl.nl
- etten-leur.nl
- eu2016.nl
- euthanasiecommissie.nl
- eutruckplatooning.com
- examenblad.nl
- excellentescholen.nl
- farmatec.nl
- festivalforensischezorg.nl
- filmfonds.nl
- fiu-nederland.nl
- flevoland.nl
- flitspanel.nl
- fondspodiumkunsten.nl
- forensischezorg.nl
- forensischinstituut.nl
- forumstandaardisatie.nl
- frtr.gov
- fryslan.frl
- functiegebouwrijksoverheid.nl
- functiemix.nl
- fvtdji.nl
- gahetna.nl
- gccs2015.com
- gddiergezondheid.nl
- gelderland.nl
- geldermalsen.nl
- geldrop-mierlo.nl
- gemeente-mill.nl
- gemeente-oldambt.nl
- gemeente-steenbergen.nl
- geschilleninstantieszorg.nl
- gewoontoegankelijk.nl
- gezondeschool.nl
- ggo-vergunningverlening.nl
- government.nl
- greendeals.nl
- groenlinks.nl
- grondkamers.nl
- hdsr.nl
- heerhugowaard.nl
- heerlen.nl
- hellendoorn.nl
- hellevoetsluis.nl
- helmond.nl
- helpdeskbouwregels.nl
- helpdesk-efactureren.nl
- helpdeskwater.nl
- hendrik-ido-ambacht.nl
- hengelo.nl
- hetcak.nl
- hetzorgverhaal.nl
- heusden.nl
- hhdelfland.nl
- hhnk.nl
- higherlevel.nl
- hillegom.nl
- hilversum.nl
- hiswa.nl
- hofvantwente.nl
- hogeraadvanadel.nl
- holanda.es
- holandaevoce.nl
- holandanomundo.nl
- holandawaanta.nl
- hollandavesen.nl
- hollandinthevalley.com
- hollandtradeandinvest.com
- hollandturkeytrade.com
- hoogeveen.nl
- hoogezand-sappemeer.nl
- hoogwaterbeschermingsprogramma.nl
- hoorn.nl
- horstaandemaas.nl
- huiselijkgeweld.nl
- humanrightstulip.nl
- hunzeenaas.nl
- huurcommissie.nl
- ibestuurcongres.nl
- ibki.nl
- ictu.nl
- idensys.nl
- ifv.nl
- igz.nl
- ilent.nl
- inburgeren.nl
- incca.org
- ind.nl
- industrialtechnologies2016.eu
- informatieberaadzorg.nl
- informatielangdurigezorg.nl
- infoterugkeer.nl
- innova58.nl
- inspectieszw.nl
- inspectievenj.nl
- integraalveilig-ho.nl
- integriteitoverheid.nl
- internationaalondernemen.nl
- internetspiegel.nl
- investingindutchhousing.nl
- investinholland.com
- iob-evaluatie.nl
- ipo.nl
- isoregister.nl
- iuc-noord.nl
- jaarberichtrvdk.nl
- justid.nl
- justis.nl
- justitieleinterventies.nl
- kabinetsformatie2017.nl
- kampen.nl
- kansspelautoriteit.nl
- kasteelgroeneveld.nl
- katwijk.nl
- kb.nl
- kbvg.nl
- kcb.nl
- kcwj.nl
- kennisnetwerkbiociden.nl
- kennisopenbaarbestuur.nl
- kennisplatformveehouderij.nl
- kerkrade.nl
- kiesbeter.nl
- kiesraad.nl
- kimnet.nl
- kinderbescherming.nl
- kinderombudsman.nl
- klimaattop2016.nl
- knaw.nl
- knmi.nl
- kombijdepolitie.nl
- koninklijkhuis.nl
- krachtontour.nl
- kustwacht.nl
- kvk.nl
- kwaliteitsafsprakenmbo.nl
- landelijkeadviescommissieplaatsing.nl
- landelijkmeldpuntzorg.nl
- leefbaarometer.nl
- leerwerkloketfriesland.nl
- lerenenwerken.nl
- letterenfonds.nl
- liec.nl
- limburg.nl
- lintjes.nl
- lochem.nl
- locov.nl
- logius.nl
- loketgentherapie.nl
- loketgezondleven.nl
- loonaangifteketen.nl
- loonopzand.nl
- losser.nl
- lvnl.nl
- lzalp.nl
- maakereenpuntvan.nl
- maakhetzeniettemakkelijk.nl
- maasdriel.nl
- maritiem-erfgoed.nl
- maror.nl
- mccg.nl
- medemblik.nl
- mediafonds.nl
- meerssen.nl
- meldknop.nl
- meldplichttelecomwet.nl
- meldpunttelecomwet.nl
- mensenrechten.nl
- mensenrechtentulp.nl
- meppel.nl
- miavamiljaarverslag2009.nl
- miavamiljaarverslag2010.nl
- middelburg.nl
- miedenproject.nl
- milieuzones.nl

- minfin.nl
- mirta2weerteindhoven.nl
- mirta67leenderheidezaarderheiken.nl
- mirtoostkantamsterdam.nl
- mirtoverzicht.nl
- mirt-rotterdamdenhaag.nl
- mjpo.nl
- moerdijk.nl
- monitorgezondheid.nl
- monumentaleinterieurs.nl
- motrainingen.nl
- multifunctionelelandbouw.net
- museumencollecties.nl
- naarnederland.nl
- nak.nl
- naktuinbouw.nl
- nationaalarchief.nl
- nationaalcoordinatorgroningen.nl
- nationaalovberaad.nl
- nationaalrapporteur.nl
- nationaleiconen.nl
- nationaleombudsman.nl
- nationaleonderwijsgids.nl
- natura2000.nl
- ncadierproevenbeleid.nl
- nctv.nl
- nederbetuwe.nl
- nederlandenu.nl
- nederlandgezondenwel.nl
- nederlandse-sportraad.nl
- nederlandsinvesteringsagentschap.nl
- nederlandwereldwijd.nl
- neerijnen.nl
- netherlandsandyou.nl
- netherlandsworldwide.nl
- nhv.nu
- niderlandy-i-vy.nl
- niederlandeweltweit.nl
- nieuwegein.nl
- nifpnet.nl
- nijmegen.nl
- niwo.nl
- niyuhelan.nl
- nji.nl
- nkca.nl
- nlintheusa.com
- nomorefoodtowaste.nl
- noorderzijlvest.nl
- noord-holland.nl
- noordzeeloket.nl
- nrgd.nl
- nvao.net
- nvwa.nl
- nwo.nl
- nza.nl
- ocwincijfers.nl
- odc-noord.nl
- oecdguidelines.nl
- oecd.org
- oesorichtlijnen.nl
- officielebekendmakingen.nl
- officiele-overheidspublicaties.nl
- omgaanmetdepressie.nl
- omgevingsloket.nl
- omgevingswetportaal.nl
- om.nl
- ondernemersplein.nl
- ondernemingsdossier.nl
- onderwijsincijfers.nl
- onderwijsinspectie.nl
- onderwijsraad.nl
- onderzoeksraadintegriteitoverheid.nl
- onderzoeksraad.nl
- onderzoeksraad.nl:443
- onehealth.nl
- onsonderwijs2032.nl
- onswater.nl
- opdrachtgeversforum.nl
- opendata-award.nl
- operatiebrp.nl
- opnieuwthuis.nl
- orandatowatashi.nl
- oteam.nl
- oude-ijsselstreek.nl
- ouder-amstel.nl
- ouders-uit-elkaar.nl
- oudewater.nl
- overbetuwe.nl
- overheid.nl
- overijssel.nl
- pagefreezer.nl
- paisesbajosmundial.nl
- paisesbajosytu.nl
- papendrecht.nl
- parismou.org
- partijvoordedieren.nl
- passendonderwijs.nl
- paysbasetvous.nl
- paysbasmondial.nl
- pbl.nl
- p-direkt.nl
- pdok.nl
- peelenmaas.nl
- permanentevertegenwoordigingen.nl
- permanentrepresentations.nl
- pianoo.nl
- pilotdtt.nl
- platformparticipatie.nl
- politie.nl
- poraad.nl
- processinnovation.nl
- proeftuinenmaakverschil.nl
- provinciegroningen.nl
- provincie-utrecht.nl
- pulsefishing.eu
- puntenstelsel.nl
- purmerend.nl
- putten.nl
- pvda.nl
- pvv.nl
- pwdji.nl
- q-bank.eu
- raadrvs.nl
- raadvanstate.nl
- raadvoorplantenrassen.nl
- raalte.nl
- randstad380kv-noordring.nl
- randstad380kv-zuidring.nl
- rda.nl
- rdw.nl
- rechtspraak.nl
- rechtwijzer.nl
- referendum-commissie.nl
- referentiegrootboekschema.nl
- regiebureau-pop.eu
- regioburgemeesters.nl
- regionaalkompas.nl
- registerleraar.nl
- rekenkamer.nl
- rekentoolfunctiemixvo.nl
- renkum.nl
- rheden.nl
- riec.nl
- rijksacademie.nl
- rijksbegroting.nl
- rijksbredekunstvoorziening.nl
- rijksdienstcn.com
- rijksfinancien.nl
- rijkshuisstijl.nl
- rijksinspecties.nl
- rijksnormering.nl
- rijksoverheid.nl
- rijksschoonmaak.nl
- rijksvastgoedbedrijf.nl
- rijkswaterstaat.nl
- rijnland.net
- rijnwaarden.nl
- rijssen-holten.nl
- rijswijk.nl
- risicokaart.nl
- risicotoolboxbodem.nl
- rivm.nl
- rivmtoolkit.nl
- rli.nl
- rob-rfv.nl
- roermond.nl
- roosendaal.nl
- rotterdam.nl
- row-minvws.nl
- royal-house.nl
- rozendaal.nl
- rsj.nl
- rucphen.nl
- ruimtevoorderivier.nl
- ruimtevoorredzaamheid.nl
- rva.nl
- rvde.nl
- rvig.nl
- rvo.nl
- rvr.org
- rwseconomie.nl
- rws.nl
- saip.nl

- samensnelinternet.nl
- samenwerkenaanontwerpkracht.nl
- samenwerkenaanriviernatuur.nl
- savewildlife.nl
- s-bb.nl
- sbr-nl.nl
- sbv-z.nl
- schadefonds.nl
- scheldestromen.nl
- schiedam.nl
- schielandendekrimpenerwaard.nl
- schoolleidersregisterpo.nl
- schoolleidersregistervo.nl
- schouwen-duiveland.nl
- scp.nl
- sectorplannen.nl
- s-hertogenbosch.nl
- sieunddieniederlande.nl
- sikb.nl
- sint-michielsgestel.nl
- skal.nl
- slachtofferinformatie-om.nl
- sliedrecht.nl
- slimmeengezondestad.nl
- slotcoordination.nl
- sluispedia.nl
- socialestabiliteit.nl
- sodm.nl
- soilpedia.nl
- spaceoffice.nl
- sp.nl
- staatsbosbeheer.nl
- staatsexamensnt2.nl
- staatvenz.nl
- stagefondszorg.nl
- startstuderen.nl
- steenwijkerland.nl
- steunpunttaalenrekenenvo.nl
- stopheling.nl
- strafrechtketen.nl
- strakszwangerworden.nl
- subsidieregeling-pg-opleidingen.nl
- svb.nl
- taalakkoord.nl
- taskforcekinderenveilig.nl
- tcbodem.nl
- techniekpact.nl
- tenboer.nl
- tenderned.nl
- terneuzen.nl
- terugvoerplicht.nl
- teylingen.nl
- thedutchchallenge.nl
- thegfce.com
- tilburg.nl
- tno.nl
- toegangocw.nl
- toekomstreligieuserfgoed.nl
- toetsingonline.nl
- toetsingscommissievp.nl
- toolkitvtv.nl
- topinkomens.nl
- topsectoren.nl
- traderouteasia.nl
- transitieautoriteitjeugd.nl
- transitiecommissiesociaaldomein.nl
- transparantiebenchmark.nl
- transplantatiestichting.nl
- trendsinbeeldocw.nl
- tubbergen.nl
- tuchtcollege-gezondheidszorg.nl
- tuinbouw.nl
- tynaarlo.nl
- ubrijk.nl
- un-psf2017.nl
- uwkindenseks.nl
- uwv.nl
- uziregister.nl
- vaarweginformatie.nl
- vallei-veluwe.nl
- varendoejesamen.nl
- vastgoedvanhetrijk.nl
- vechtstromen.nl
- veiliginternetten.nl
- verkeersonderneming.nl
- verkiezingsuitslagen.nl
- verlofadviescollege.nl
- verruijt.net
- veva.nl
- videnet.nl
- volksgezondheidenzorg.info
- vraagbaakiv3gemeenten.nl
- vraaghetdepolitie.nl
- vvd.nl
- vwscongresmagazine.nl
- waarderingskamer.nl
- waddensea-worldheritage.org
- waterandthedutch.com
- waternet.nl
- waterschaprivierenland.nl
- wdodelta.nl
- weekvanhetgeld.nl
- wegwijzermensenhandel.nl
- welvaartenleefomgeving.nl
- werkenbijdeeu.nl
- werkenbijdefensie.nl
- werkenvoorinternationaleorganisaties.nl
- werkenvoornederland.nl
- werk.nl
- wetterskipfryslan.nl
- wijzeringeldzaken.nl
- wijzeringeldzaken.nl:443
- windenergie.nl
- wlo2015.nl
- wmowerkplaatsen.nl
- woningwet2015.nl
- won-nl.org
- wrij.nl
- wrr.nl
- wshd.nl
- wtzi.nl
- zeeland.nl
- zeeweringen.nl
- zelfinspectie.nl
- zeteenstreepdoordiscriminatie.nl
- zichtopdevreemdelingenketen.nl
- zonmw.nl
- zorgcsp.nl
- zorginstituutnederland.nl
- zorgopdekaart.nl
- zorgpact.nl
- zorgvoorinnoveren.nl
- zovar.nl
- zuiderzeeland.nl
- zuid-holland.nl
- zuid-west380kv.nl