Forensic Profiling Of An eBook Reader: An Example

M. Piccinelli and P. Gubian

Dept. of Information Engineering, Faculty of Engineering,
University of Brescia, Italy
email: mario.piccinelli@ing.unibs.it

Abstract

Forensics profiling refers to the study and exploitation of traces in order to draw a profile relevant to the investigation about criminal or litigious activities. While traces may not be strictly dedicated to a court use, they may increase knowledge of the subject under investigation. In this context we will study the evidence found in a modern ebook reader, and we will explain how it could be used during an investigation to help understand the profile and the habits of its owner by building a reliable timeline of all the interactions between the user and the device. We use as an example a modern ebook reader, the Sony Touch PRS-650, of which we present a complete profiling made with custom software.

Keywords

Ebook Reader, Forensics, Profiling.

1. Introduction

The aim of forensic research is to support investigatory and judicial processes by finding traces in otherwise apparently unpromising raw material from which it is possible to build a picture of events and activities (WP6, FIDIS Consortium, 2008). But digital evidence is not well perceived by human senses (Wang, 2007): crucial pieces of digital evidence may simply be overlooked due to the fact that examiners do not fully comprehend how seemingly useless pieces of data can be converted to evidence of high value (Koen R.). If, thanks to this apparently worthless information, an investigation team can understand an intruder's modus operandi, it might be possible to determine various attributes describing the intruder, such as skill level, knowledge and location (Casey, 2002). In this paper we decided to analyze an ebook reader. This kind of portable electronic device is seeing a notable growth, and it is becoming common for a forensics examiner to find such a device during an investigation. What we found during our research is that there is not a common methodology for the examination of this kind of device, mainly because readers contain a small amount of personal information regarding their owner and thus are deemed of little interest in an investigation. We will describe a new source of information: the time data that can be acquired from the logs of an ebook reader. This kind of evidence, which may be perceived as worthless, could in fact be helpful for acquiring additional information about the user of the device itself.

2. Ebook readers

Ebook readers are portable electronic devices designed primarily for the purpose of reading digital books. These devices, similar in form to tablet computers, are usually provided with hardware and software developed specifically for this single task. Each of these devices is considerably different from each other, in terms of hardware characteristics, connectivity (some models even ship with built-in WiFi connectivity) and data storage, and thus it is impossible to create a standard examination protocol. In this paper we will focus on a single model, the Sony Reader PRS-650.

2.1. Example: Sony Reader Touch PRS-650

The Reader Touch PRS-650 is a modern ebook reader manufactured by Sony. It uses an electronic paper display (6 inch, 16-level gray scale, 800x600 pixels), has a tablet form factor and is powered by a Lithium-ion rechargeable battery. The touch screen represents the main input; 5 buttons on the front face provide other input. It runs on MontaVista Linux and provides 2 GB of internal flash memory, which can be extended by a removable SDHC card and/or a removable Memory Stick PRO duo card. This ebook reader provides the capability of reading books in various formats and taking notes (handwritten on the touchscreen or typed into a simulated keyboard); it also provides instruments for saving bookmarks and drawing or highlighting words on each page of books (Sony Corporation, 2011). These capabilities will be explained in detail later as they will be the main sources of data for the profiling of the device.

3. Data Acquisition

Sony provides a client software named "Sony ebook library" to interact with the reader from a personal computer with Windows or Mac OS X operating systems. Nonetheless, the reader's storage can also be accessed without using specific software because the device itself is recognized, when connected to a PC via a standard USB cable, as a standard USB mass storage device. In detail, when the device is connected to a PC, it shows two to four single partitions:

1. a partition labeled "SETTING", formatted as MS-DOS FAT16, of 10.4 MB (of which 3.8 available) containing the installers for the Windows and Mac OS X versions of the Sony ebook library;

2. a partition labeled "READER", formatted as MS-DOS FAT32, of 1.61 GB containing the books stored in the internal memory of the device along with two folders, "database" and "Digital editions". The content of these two folders will be described later, as they contain the data we will use to build a profile of the user of the device;

3. if available, also the memory cards in the devices will be mounted as single partitions with their real volume name. These will contain, along with the

books stored in them, two folders named "Sony Reader" and "Digital Editions". Also these folders will be described later.

All the data contained in these volumes can easily be acquired and analyzed with standard forensically sound methods (see Carrier, 2005). It is noteworthy that this model isn't capable of wireless connectivity such as WiFi or 3G, which makes the forensics examination easier by rendering the device insulated from the outside world (and thus tamper proof) except for the USB connection, which can be write-protected by hardware or software means.

4. Data structure

As mentioned above, the analysis is based on the data found in specific folders in the internal storage of the reader and in the removable storage devices. During our research we found that the most interesting data are stored in the folders named *database* (on the internal storage) and *Sony Reader* (in the removable storage).

4.1. Sony Reader folder

The structure of a sample *Sony Reader* folder found in the SD card of the device under test is shown in Figure 1. The content of the single folders deemed useful for our analysis is described below.

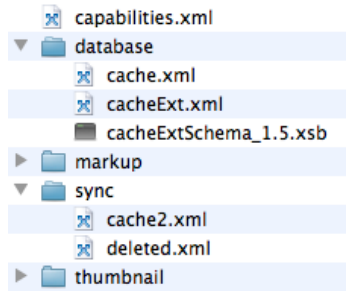

Figure 1: *Sony Reader* directory structure

The file *capabilities.xml* stores information about the reader device: a list of the supported file types, a list of the supported markup elements, the display resolution and so on.

The folder *markup* contains the graphical files depicting the hands-free markups drawn on the eBooks by using the provided pen on the touchscreen. This directory mimics the directory tree of the SD card, each branch terminating with a last folder with the same name as the ebook it references. Each one of these low level folders contains the markup files for a single book. For each markup we found two files, with the same name (the unique id of the markup, as described later) but different extensions.

1. a JPG file, a compressed bitmap depicting a preview of the book page with the markup. Each JPG file weighs about 4 to 8 KB and has a resolution of approximately 110 x 150 pixels.

2. an SVG file, a vector graphics file depicting the markup drawing.

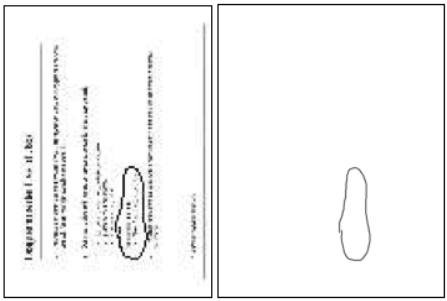

Figure 2: Example of hands free thumbnail and vector graphic

The folder *thumbnail* contains a directory structure similar to the one described for the *markup* folder. Each lowest level folder contains a single graphical file named *main_thumbnail.jpg* depicting a thumbnail representation of the book it references (usually the book cover).

The folder *cache* contains two XML files containing the main data used in our analysis, *cache.xml* and *cacheExt.xml*. The file *cache.xml* stores a list of all the books in the SD card. Each element of the list is saved as an XML node named "text", with attributes describing the document it refers to such as book title and author name, full path of the file relative to the root of the SD card, last page read and part (as the page can be zoomed and shown in more parts), a unique numerical id, MIME type of the file, date of creation, size in bytes. The file may also contain XML nodes named *playlist* describing lists of books; each node contains one child node for each book in the collection, identified by its unique id.

The file *cacheExt.xml* also stores a list of all the books in the SD card. Each element of the list is saved as an XML node named *text* with an attribute *path* containing the full path of the referred document relative to the root of the SD card. Each node may contain several child nodes:

1. *currentPosition*: a node depicting the position of the last read page of the document (page number and part) along with a timestamp. The node itself may contain a *mark* node depicting the current position in a support-dependent form, encoded in base64 format. For PDF files this position is represented as Adobe #pdfloc coordinates, (such as "#pdfloc(c81e,26)") while for epub files this position is represented as a link to a point in the

appropriate xhtml file inside the epub structure (such as "index_split_4.xhtml#point(/1/3/146/2/1/1:391)").

2. *preferences*: among the others, the user selected values of brightness and contrast. This node may also contain a child node of type *dicHistories*; each child of this node (of type *dicHistory*) is a recording for the action of looking up a word in one of the built-in dictionaries, which is achieved in the device under test by double clicking on the word while reading the document. Among the attributes of these bottom level nodes are the word looked up and the timestamp of the operation.

3. *thumbnail*: a node referring to the thumbnail image file for the book (specifying width and height), stored in the *thumbnails* folder seen above.

4. *history*: a node listing the last pages shown by the device (at most 100 entries). For each page seen, an *item* XML node is created with attributes describing, among other things, the page read (page number and part) and the timestamp. This node also contains two child nodes, *comment* and *mark*, containing the same information seen for the *currentPosition* node.

5. *markups* and *deletedMarkups*: these two nodes contain the markups and the deleted markups respectively for the document they refer to. These markups are stored as child nodes of three types: *annotation*, *freehand* and *bookmark2*. An annotation represents a piece of text that has been highlighted on the device; a freehand represents an annotation drawn on the device using the provided pen on the touchscreen; a bookmark represents a page (or a part of it) bookmarked. Each of these node types has different characteristics (for example, annotations store two child nodes named *start* and *end* pointing to the position of the selected text in support dependent format, as seen before for the *mark* nodes, while freehands also store child nodes pointing to the SVG file of the annotation and the thumbnail file of the page showing the drawing) but they all have some attributes in common, the most interesting of them being the position in the document (page number and part) and a timestamp (of creation for markups or deletion for the deleted ones) which will be used for building the timeline.

Notice that the child nodes of many of the nodes described before are related to an action performed by the user (moving to a page of a document, adding a bookmark, drawing a freehand note) and are provided with a timestamp, showing the date and time when the action was performed. These timestamps will be used later to build a timeline of the use of the device.

4.2. Database folder

The *database* folder found in the main storage (depicted in Figure 3) of the device corresponds the *Sony Reader* folder found on the SD card, with some remarkable exceptions.

The most easily detectable exceptions are that by default this folder also contains a subfolder named *media*, in which we found the multimedia files that came preloaded in the device and the files for the note application:

1. audio folder, containing some MP3 files;

2. books folder, containing some books in EPUB and RTF format;

3. images folder, containing some JPG images (by default they are used as screensavers);

4. notepads folder.

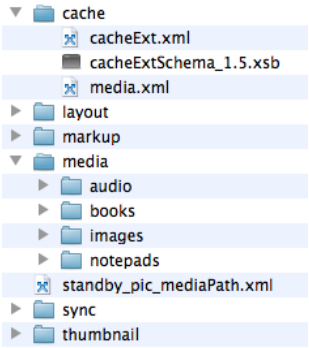

Figure 3: *Database* directory structure

The latter is the most useful one from a forensics perspective because it contains the files created by the note application. The note application lets the user create hands free notes (similar to the hands free markups which can be drawn on book pages), and stores them as single files in this folder with a *.note* extension. Each note file is in fact a vector graphic file, stored in an XML format similar to the SVG type, in which each line composing the drawing is described as a list of X/Y coordinates. The XML head of the file also contains a *note:notepad* attribute named *createDate*, which holds the creation date of the note file in Unix epoch format, which will be combined with other timing data to create a timeline of the device use.

Another difference between the *Sony Reader* folder described before and the *database* folder is that the latter contains, instead of the *cache.xml* file, another XML file named *media.xml* with a different structure. For documents, the *cache* nodes seen in *cache.xml* are replaced by *cache:text* nodes, holding the same information as seen before. Other node types are provided to support the different media types held in the device main storage:

1. *cache:notepad* nodes holding data about the .note files described before; among the attributes found in these nodes the most interesting are the creation timestamp and the full path of the .note file.

2. *cache:audio* nodes holding data about the MP3 files found on the device.

3. *cache:image* nodes holding data about the image files stored on the device.

4. Other node types deemed not useful for a forensics analysis: *cache:library*, *cache:watchSpecial*, *cache:playlist*.

All the aforementioned nodes are contained under a parent node named *records*.

An *cacheExt.xml* file was also found also in the *media* directory with a structure identical to what described for the *Sony Reader* directory. The only difference found is the presence of other node types different from *text*, used to describe documents; these nodes, whose types are *audio*, *image* and *notepad*, refer to the other media types described before but don't seem to hold forensically useful data.

5. Profiling with the data found

As we have shown in the previous section, during normal interaction between the user and the device a vast amount of data is created in the *Sony Reader* and *database* directories. This amount of data, stored as nodes in XML files, can be easily extracted from the device itself or an image of the device mass storage and attached cards created by standard forensically sound tools. During the extraction the data can also be easily protected from tampering, for example by producing an MD5 hash signature of each file upon extraction and checking it against an hash of the same files on the device.

During this research we focused on the temporal data. While all the content of the device and its attached storages can be useful during a forensics examination, standard forensics tools can analyze those elements. The most interesting evidence we found that is not common to other digital devices is the presence in the cache files of timestamps for each action performed on the device, such as:

1. last reading of a document;

2. creation date of a document (as held by the filesystem);

3. creation date of a note;

4. reading of a page of a document (at most the last 100 for each document);

5. creation and deletion of markups (bookmarks, free hand drawings and highlighting);

6. look up for words in the built in dictionaries.

With all this data we will be able to create a chronology of the use of the device. For this purpose we created a simple Python script, which parses all of the nodes

described above to acquire their timestamp and then outputs them in chronological order.

All the timestamps found in the device are stored as strings in GMT timezone, like "Thu, 18 Aug 2011 19:49:30 GMT". Unfortunately, the effective timezone of the device is not stored in the files described before nor anywhere else in the readable partitions, and it can't even be seen on the device settings menu, as it is set when the device in synchronized with the Sony software and cannot be modified by the user. The only way we found to know the offset between the device time and the local time is to compare the creation dates of the files, as written in the file "*cache.xml*" and as provided by the filesystem. This method, though, hasn't proven always reliable, especially if the device was frequently updated by adding files from different host computers and different applications, because even third-party software such as *Calibre* can't reliably identify the timezone setting of the device. A way to have these elements cleaned is by inserting in the device a clean SD card with just one file in it, and let the reader create its "*cache.xml*" file which will contain a timestamp to check against the real creation time of the file itself. This way the investigator has all means to know the real timezone of the device and the difference in time between the device itself and the reference time.

5.1. Analysis of sample data

To study the amount of data created during normal use of the device, we proceeded by creating simple interactions with a brand new document we uploaded for this purpose; after each interaction we extracted the XML files from the device and analyzed them.

We uploaded the test file (test.pdf) on the SD card by connecting the device with a personal computer with Mac Os X 10.6 Snow Leopard and copying it to the mounted volume corresponding to the card itself. After that we disconnected the device and let it restart (the device, when disconnected from the host computer, analyzes all its stored documents and updates the configuration files). Then we ran our utility and saw that the only element found by searching for the file name is the record in *cache.xml* with the creation date of the document, which is equivalent to the creation date of the file as seen by the host computer filesystem.

```
2008-12-22 17:28:04    Creation date of test.pdf
```

After that we just opened the book (without browsing any page) and repeated the analysis. This time the device added to the *cache.xml* file the current position (page 0) and added to *cacheExt.xml* a history element depicting the reading of page 0.

```
2011-09-29 13:18:24    Current position page 0 of 291 of test.pdf
2011-09-29 13:18:24    Reading page 0 of 291 of test.pdf
```

The next step was browsing through the first pages of the book, and repeating the analysis. This time the device had correctly recorded as history elements all the pages we opened in the rightful sequence. It is interesting to note that the first time

the cover page was opened it was recorded as a 0 page, while after turning to the next page it saved two records, one for the second page (as expected) and one for the first page (with the same timestamp). Also, the current position data was updated with the new timestamp and new page number.

```
2011-09-29 13:18:24   Reading page 0 of 291 (offset 0) of test.pdf
2011-09-29 13:20:13   Reading page 2 of 291 (offset 0) of test.pdf
[...]
2011-09-29 13:20:37   Reading page 9 of 291 (offset 0) of test.pdf
2011-09-29 13:20:39   Current position page 9 of 291 of test.pdf
```

After that we created some markups: a bookmark, a free hand drawing and a highlighting. The analysis correctly reported all these operations. Also, the file *cache.xml* was updated with a bookmark record with the timestamp of the last markup annotation.

```
2011-09-29 13:22:26   Bookmark2 markup at page 9 of 291 of test.pdf
2011-09-29 13:22:59   Freehand markup (1317302575979.063.svg) at page
       9 of 291 of test.pdf
2011-09-29 13:23:18   Bookmark date of book ottimismo.pdf
2011-09-29 13:23:18   Annotation ("ont") at page 9 of 291 of test.pdf
```

At last, we proceeded with deleting the markups and creating some new bookmarks, to see how the system managed the deleted markups. As we can see, the entries for the markups described before were deleted, and new entries were added under the *deletedMarkups* node in *cacheExt.xml*, with the timestamp of the deletion date (this means that we lost the creation date of the markups during the process).

```
2011-09-29 13:26:46   Deleted Bookmark2 at page 9 of 291 of test.pdf
2011-09-29 13:27:23   Deleted Annotation at page 9 of 291 of test.pdf
2011-09-29 13:27:36   Deleted Freehand at page 9 of 291 of test.pdf
[...]
2011-09-29 13:28:23   Bookmark2 markup at page 12 of 291 of test.pdf
2011-09-29 13:29:08   Bookmark2 markup at page 13 of 291 of test.pdf
2011-09-29 13:29:09   Bookmark date of test.pdf
2011-09-29 13:29:25   Current position page 13 of 291 of test.pdf
```

With the results from this last test we built a custom script to create a graph with the usage data of the device (Figure 4), limited to the useful selected time span and excluding the file creation date (which is not related to an interaction between the user and the device and could be misleading). In the graph, each line represents an interaction (we didn't find useful to discriminate among different kinds of interaction). This way we can easily understand how the device was used during this time.

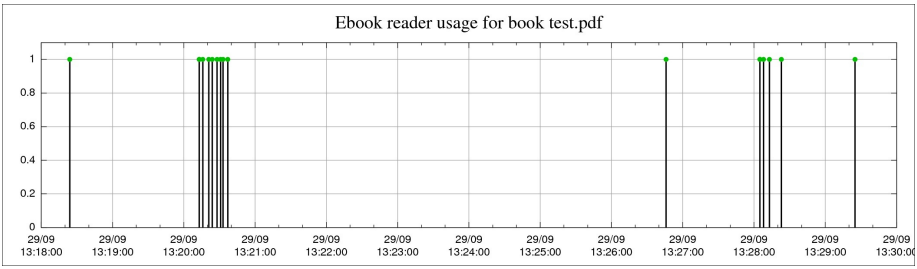

Figure 4: Usage data related to the test described before

We also performed other more general tests on the device, and obtained some more useful usage graphs related to a longer time span. For example, Figure 5 shows the usage of the device in a two months time span. In this graph we chose to differentiate among interactions regarding different books, by assigning each book a different value on the Y-axis. This way we can also see whether the user accessed many times the same book or instead briefly accessed many different documents.

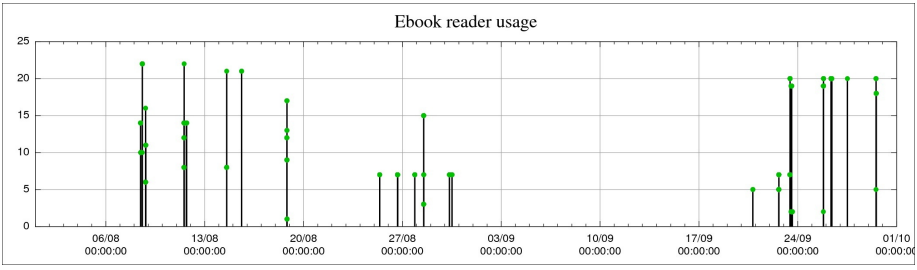

Figure 5: Usage data related to a 2 months timespan (august-september 2011)

6. Conclusions

The analysis performed on the device confirmed that it is possible to extract a forensically sound timeline of the usage of the device itself, and this timeline proved to be accurate at the single second. Virtually each operation performed by the user is logged and can be added to the timeline. During an investigation the timeline can be associated with a suspected, provided that investigators can prove that the device has always been in the exclusive availability of a single person. The evidence gathered this way could then be used in court to draw a behavioral profile of a suspected offender, support or deny an alibi, or provide additional useful information about the owner of the device.

7. About the software

The software used during this analysis was written by the authors and named “Sony Ebook Reader Time Profiler”. It is a Python script and works from the command line, without graphical interface. When the application is launched, it scans the

provided path/paths searching for files named *media.xml*, *cache.xml* or *cacheExt.xml*, and parses each one searching for timestamp data. This data is then ordered and printed on the console. Additionally the software can also build a data file that can then be fed to a provided GnuPlot script to create a timeline graph.

The code has been released under an open source license and is available at the following URL: *https://github.com/PicciMario/Sony-Ebook-Reader-Time-Profiler/.*

We encourage interested people to try the software and maybe participate to the development by providing suggestions or code to handle different devices or reporting bugs.

8. References

Wang, S.-J. "Measures of retaining digital evidence to prosecute computer-based cyber-crimes." *Comput. Stand. Interfaces* 29, no. 2 (2007): 216-223.

WP6, FIDIS Consortium. *D6.7c: Forensics Profiling.* FIDIS Consortium, 2008.

Casey, E. "Uncertainty, and loss in digital evidence." *International Journal of Digital Evidence* 1, no. 2 (2002).

Carrier, B. *File System Forensic Analysis.* Addison Wesley Professional, 2005.

Koen R., Olivier M. S. "The use of file timestamps in digital forensics." ICSA, University of Pretoria, South-Africa.

Sony Corporation. *Sony Ebook reader PRS 650.* 2011. http://www.sony.co.uk/product/rd-reader-ebook/prs-650 (accessed 2011).