2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)

CyberPDF: Smart and Secure Coordinate-based Automated Health PDF Data Batch Extraction

Reza M. Parizi
Department of Software Engineering and Game Development
Kennesaw State University
Marietta, GA, USA
rparizi1@kennesaw.edu

Liang Guo
School of Engineering and Computing Sciences
New York Institute of Technology
NYC, NY, USA
llguo06@nyit.edu

Yao Bian
School of Engineering and Computing Sciences
New York Institute of Technology
NYC, NY, USA
ybian@nyit.edu

Ali Dehghantanha
Security of Advanced Systems Lab
University of Guelph
Ontario, Canada
adehghan@uoguelph

Amin Azmoodeh
Department of Computer Science & Engineering
Shiraz University
Shiraz, Iran
azmoodeh@cse.shirazu.ac.ir

Kim-Kwang Raymond Choo
Department of Information Systems and Cyber Security
University of Texas at San Antonio
San Antonio, Texas, USA
raymond.choo@fulbrightmail.org

ABSTRACT

Data extraction from files is a prevalent activity in today's electronic health record systems which can be laborious. When document analysis is repetitive (e.g., processing a series of files with the same layout and extraction requirements), relying on data-entry staff to manually perform such tasks is costly and highly insecure. Particularly analyzing a large list of PDF files (as a widely used format) to extract specific data and migrate them to other destinations for later use is both tedious and frustrating to do manually. This paper addresses a very practical requirement of batch extracting data from PDF files in health data document analysis and beyond. Specifically, we propose a Coordinate Based Information Extraction System (CBIES) to instrument a smart and automatic PDF batch data extraction tool, releasing health organizations from duplicate efforts and reducing labor costs. The proposed technique enables users to query a representative PDF document and extract the same data from a series of files in the batch analysis manner swiftly. Furthermore, since security and privacy considerations are essential part of any health record systems, it is included in our approach. Based on CBIES, we implement a prototype tool for PDF batch data extraction technique named, CyberPDF. The tool exhibits great efficiency, security and accuracy in multi-file data processing.

CCS CONCEPTS

• **Security and privacy** → *Software security engineering*;

KEYWORDS

Document analysis, Data extraction, Secure health record aystems, PDF data, Batch processing, Software tool support.

ACM Reference Format:
Reza M. Parizi, Liang Guo, Yao Bian, Ali Dehghantanha, Amin Azmoodeh, and Kim-Kwang Raymond Choo. 2018. CyberPDF: Smart and Secure Coordinate-based Automated Health PDF Data Batch Extraction. In *ACM/IEEE International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE '18), September 26–28, 2018, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3278576.3281274

CHASE '18, September 26–28, 2018, Washington, DC, USA

ACM ISBN 978-1-4503-5958-0/18/09.
https://doi.org/10.1145/3278576.3281274

1 INTRODUCTION

Portable Document Format (PDF) is widely used by individuals and organizations [1], including in the healthcare sector. One of the most important demands is to automatically extract and transfer data from medical PDF files (e.g., bill invoices, patient data, or prescriptions) in a batch and migrate them to the appropriate format (e.g., Microsoft Excel or Access) or even third-party system. Currently, most companies or businesses deal with this kind of data processing in an ad-hoc, costly and obsolete way, which is relying on manual labor.

On the other hand, security and privacy are a topical subject in health record systems [22–24]. Furthermore, PDf is a highly vulnerable format to inject malicious code [25] and transferring it through a medical software system is a real threat to data confidentiality, integrity and availability.

After studying related literature and current products, we found a Free and Open-Source Software (FOSS) automated tool is currently scarce in the market and literature, especially within the healthcare document analysis domain. Hence, this research was inspired to take the first step to alleviate such problem by proposing and developing an automated solution and its tool support that facilitates the batch extraction of data from PDFs.

As part of the contributions, the proposed tool offers the following distinguished characteristics: (1) The tool is both smart and fully automated with a straightforward flow. (2) Besides, the tool is lightweight and efficient. Processing a batch of x PDFs requires

less than a few seconds without causing any major overhead. (3) Security and privacy principles are considered in design of the tool.

The rest of this paper is structured as follows: Section 2 presents related work and background. Section 3 gives the details of the system and Section 4 describes the CyberPDF tool's implementation. Section 5 gives the conclusion of our work.

2 RELATED WORK

Information extraction is a very common task in all areas of businesses. It is the task of automatically extracting structured data from unstructured or semi-structured machine-readable documents. The administrative part of businesses more often needs to shift through a lot of PDFs' data in their daily operational fashion so that they serve the consumers better. As a result, a number of data-entry employees will be assigned to manually analyze single-by-single PDFs for data extraction and entry, bringing a lot of unnecessary expenses to the organization [5].

Ramakrishnan et al. [6] proposed an approach that detects information using a layout-aware text extraction technique from full-text PDF of scientific articles. Their proposed work was mainly designed for developers of biomedical text mining or biocuration informatics systems that use published literature as an information source. The other work reported in [7] simply converts PDF to HTML based on a Text Detection approach. Though this work has provided a workable method to convert information from PDF files, it fails to propose any batch-based data extraction feature nor data migration.

Aiello et al. [8] proposed a document analysis system to assign logical labels and extract the reading order in a broad set of documents. The main focus in this work is on the analysis of heterogeneous collections of documents based on generic knowledge. Similarly, Bart and Sarkar [9] proposed a general method for extracting repeated structure from document images. According to the authors, the main novelty of their proposed approach is in formulating a probabilistic framework for extracting such structure. Within document images-based approaches, the work proposed in [10] addresses a client-driven approach to automatically extract information content within the tables in document images. Their underlying technique uses a graph-based representation of a set of key-fields selected by clients and performs graph mining in a document to produce a model. The produced model is then used to extract information content in the absence of clients.

In a more simpler work, Hassan and Baumgartner [11] proposed a flexible method for detecting and understanding tables in PDF files. In their technique, they transform the low-level PDF elements into text segments, lines and boxes on a page. From there, they convert the identified tabular presentations into HTML for information extraction purposes.

All of the above studies fall short in addressing batch processing of files (they are based on single-processed manner), and the convenience of automatic extraction with multiple data points. It is evident that the batch-processing feature of our work is worthy of use.

On the other hand, a number of solutions such as *NITRO* [12], *CogniView PDF2XL* [13], *Adobe Acrobat Pro.* [14], *A-PDF* [15] and *Tabula* [16] have demonstrated PDF-centric data extraction capabilities based on various products to create workable tools so that the businesses and individuals can use. To the best of our knowledge, there is no tool existing in the market that has batch-strength data extraction capability over a large number of PDFs.

3 THE PROPOSED SYSTEM

In this section, we introduce the basic concepts, principles of smart, automatic data processing mechanism and propose the system model, Coordinate Based Information Extraction System (CBIES) to structuralize the whole technological process (presented in Figure 1).

The system, as a smart data extractor, firstly should know the position of user's interest data (PoI). Since the user wants to batch extract data from a series of files, the system should ask the user to select one representative file (out of the batch), or alternatively displays one if the user has no particular preference. Either way, user starts selecting PoIs. Once this is done, the system also needs to know the aim position (AP), which is a specific location in a destination file (such as a cell in an Excel file) that will be used as a starting point to store selected data from PDFs (through *File Content Viewer*). These two pieces of input values will get converted behind the scenes to corresponding description parameters according to the user's operating system and current window's size through *Graphical PoIs Parameters Converter* technique. The converting results are further stored in a special intermediary file, called the "description file" that holds the coordinates of all selected data. Then, the tool uses this description file as a profile of Data *Extraction and Forward Engine* to batch extract from all the files. Our said coordinate based information extraction system consists of three main parts, as shown in Figure 1, and are described in more details in Section 3.1 through Section 3.3.

3.1 File Content Viewer

The file content viewer is the graphical user interface between users and the proposed system. In addition to other functionalities, it mainly provides a visual edge to display the file content of user's choice, so that the user can navigate through the file content and identify PoIs.

With the help of graphical user interface, user can easily determine PoIs, through selecting rectangular regions, using mouse or touch screen (technically, a PoI is a rectangle region chosen graphically by the user that contains a piece of data to be extracted). The coordinate axis on the windows of the file content viewer is used to describe the position of arbitrary points (An example of file content viewer's page and PoI is shown in Figure 2). Nevertheless, a user may select more than one PoI on a representative file from the batch. The information of all selected rectangle regions will be stored in the description file, which will serve as the profile to batch process of all files with the same layout and extraction requirements. Simply put, the description file stores the coordinates of those rectangle regions, selected PoIs. The description file stores the coordinates of those rectangle regions, selected PoIs. When the user selects PoIs, the corresponding aim positions (APs) should be also inputted and stored in the description file. APs can be arbitrary

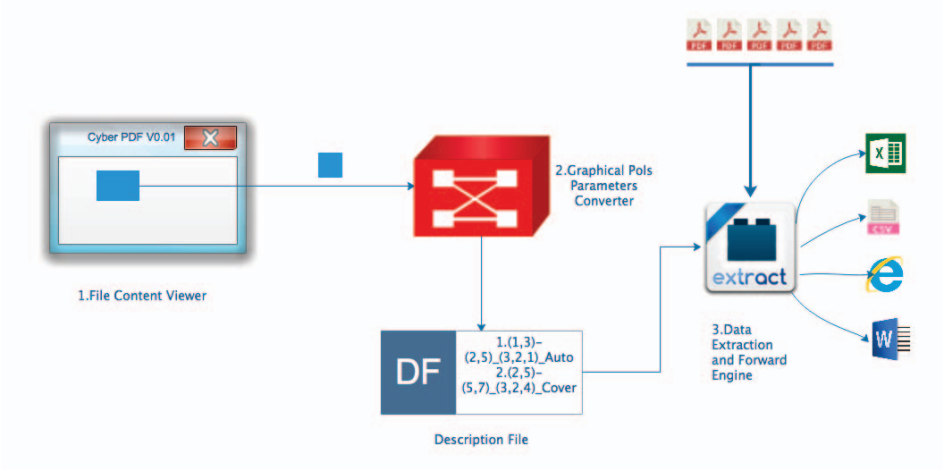

Figure 1: Conceptual model of CBIES

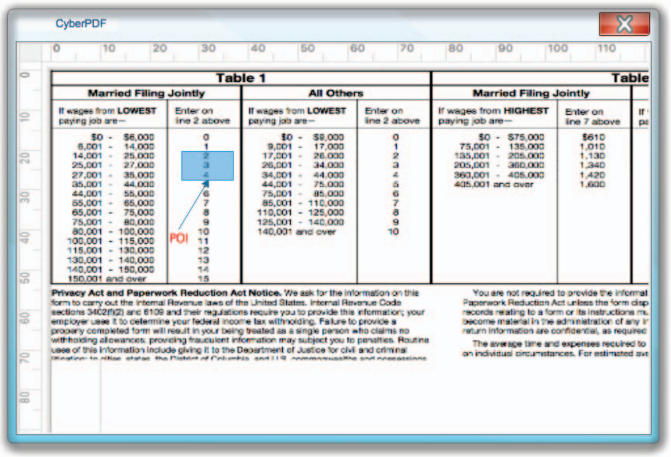

Figure 2: An example of a PoI on the file content viewer's GUI

positions in the various file formats, including Excel, CSV, HTML, Word. A simple and common aim position used in this paper is a cell in an Excel format.

3.2 Graphical PoIs Parameters Converter (Relative Coordinates Converter)

As mentioned earlier, the PoIs (rectangle regions) indicated by the user cannot be used directly to feed the system. It is not hard to understand. For example, when the user opens the representative PDF, the first page is displayed on the screen, he selects one PoI, assuming its coordinate is from (1,1) to (4,3), then he wants to choose another PoI underneath the first one, but he cannot see the text detail, so he zooms out the page and selects that PoI. When user zooms out or zooms in the page, the coordinate of each element in the page get changed. It is possible that the new PoI's coordinate is still from (1,1) to (4,3), but it would be quite unpredictable.

This is a very serious problem, and since our system needs to ensure the extracted data are accurate, with no element missing or redundant, we will have to convert the graphical coordinates to fixed coordinates relative to the raw PDF page.

When we display the PDF page, we put the page one on the panel of software. Assuming the left upper vertex is the origin of coordinates, then building a coordinate system (Java uses this kind of coordinate system), displaying the page from the left upper vertex of the panel. In other words, we put the page in a coordinate system. It needs to be emphasized that this coordinate system is fixed, no matter if the page is zoomed in or out.

The simplest way to get the fixed coordinate relative to the raw PDF page is displaying the PDF page with its raw size. The common size of the PDF page is A4 and it is easy to be displayed on computer screens.

We tend to use raw page's size to display, as this is the most accurate and simple way. While we had to consider the requirement of zooming out and in, the wise choice is to use the same scale factor to enlarge and shrink the length and width of the page. We define the scale factor as S. Pick a point on the page, define the coordinate relative to the raw page as (X , Y), the zoomed in or out coordinate as (X', Y'). The formula used to get (X, Y), is defined as follows:

$$Coordinate \begin{cases} X = X'/S \\ Y = Y'/S \end{cases} \quad (1)$$

Considering Figure 3 as an example, assume the raw size is 40×20, enlarge the page with S=2. The after size is 80×40. Pick a point A', its coordinate is (80,40), according to Formula (1) we can get the relative coordinate.

3.3 Data Extraction and Forward Engine

There are few program libraries (APIs) related to PDF processing. One of the best well-known APIs, is Apache PDFBox [17] that we have adopted in our work as part of the core components for data extraction process. It is based on Java technology and can extract data from a specified region in the PDF [18]. The region is basically a rectangle and is described by a two-point coordinate. For example,

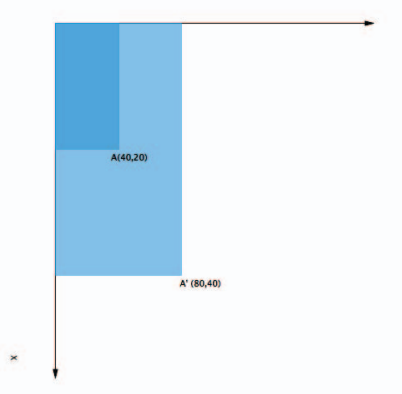

Figure 3: Raw page and Enlarged page

we can use coordinate (1,1) and (4,3) to determine a rectangle with length 3, width 2.

Though we have taken PDF as a starting base in this current work, the whole process also applies to other file formats. It is no doubt PDF is one of the most popular formats despite of nations, businesses, or individuals. We take PDF process component as the main part of the Data Extraction Engine, while we also consider other formats, for example the images (JPG, PNG etc.), which can use OCR technology to process.

The best way to stretch the applicability and portability our work to support most of the file formats is to design the Data Extraction Engine in a modularized way. This aspect was achieved by adopting a pluggable architecture. No matter what kind of data extraction component we use, it can be easily added to the system's architecture [27] without compromising the flow of the process. In such circumstances, all the component in place needs to do is to use the two input values processed by *Graphical PoIs Parameters Converter* (introduced in Section 3.1) to get the fixed parameters and use those for extracting user's data.

In the context of extraction process, another problem that we needed to solve was how to forward the extracted data. In other words, we needed some intermediary tools to help us read and write a specific file to store data. Since most of companies and individuals use MS Excel to process data, it was then chosen as the first format supported by our system. Many program libraries in the market provide the interface of processing Excel. One of the most suitable is Apache POI [19].

4 CYBERPDF TOOL

Built upon CBIES, we designed an automated PDF data extraction software, and implemented a prototype tool named, CyberPDF. This section describes the tool's implementation. The preliminary version of the tool is available at: https://github.com/LeonKwok0/CyberPDF.

4.1 Technology stack

We used Java-inspired technology to implement this application with the help of two Java based well-known APIs, Apache PDFBox and Apache POI. The current version of the developed prototype is a desktop app that can run in server platform, Windows/MacOS/Linux. Its GUI is straightforward and the look and feel is Microsoft office style.

4.2 Operational Workflow

CyberPDF is simple, yet efficient to use and it is designed in a three-step workflow.The *first step* is to either create a new extraction project or resume an existing project. In case of new creation, the tool helps user to choose a rep PDF from a series of PDFs with the same layout and extraction requirements (simply, choose one from the batch). This PDF file is considered as the working sample to help the user determines the data that she/he wishes to extract on the entire batch. The sample PDF will be shown in the file content viewer of the tool and the user gets ready to select PoIs on it. The *second step* is to select PoIs and form the description file by simply clicking the Start Button and using the mouse pointer to select the data. During this process, a dialog window prompts the user to enter the aim position (AP) in the Excel file. After selecting the desired PoIs, the user can click the Save button to save those PoIs and AP in the description file. This step is simple enough that the user just needs, selecting and entering a tiny information (as shown in Figure 4 [Left Window]).

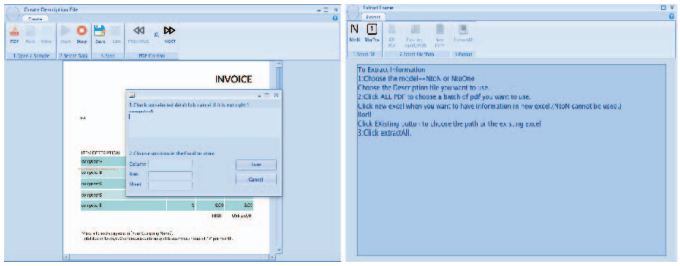

Figure 4: Left Window :Main panel, select PoIs & Right Window: Extract Panel

After the user saved the information in the description file, the tool asks whether he/she wants to go to the next step, i.e., extracting data from PDFs (a.k.a the batch). If the user decides to extract data at that moment he/she will click OK to proceed. If he/she decides to postpone the job to another time, he/she can click the Exit Project button, and resume the extraction part at a later time by retrieving the created description file without the need to repeat the first two steps. The *third step* is to extract data from the series of PDFs (Figure 4[Right Window]). At first, the user needs to choose the description file (in case of resuming a previous attempt). Meanwhile, the model of storing the extracted data is also chosen. As of now, CyberPDF offers two data storage models. One is extracting data from multi-PDFs and store them all in one Excel file, row after row (i.e. N-One). Another one is extracting data from one PDF and store the data in one Excel file, repeating this process until all the PDFs are processed (i.e. N-N). At last, the user simply needs to assign the

path of existing Excel file or assign path and name to a new Excel file. Done! In the process of extracting, a dialog pops up if the tool detects there are existing data on the selected destination in the Excel file, giving the user options of either overwriting the existed data or use a next row to land the data.

4.3 Features

Language Compatibility. In the world, individuals and companies use PDFs in various languages to handle their daily works or business operations. Thus, it is very important that CyberPDF can be compatible and workable in different language environments. Because of the architecture and the internal APIs that are used by CyberPDF, the tool can support all the mainstream languages, including but not limited to English, Chinese, Japanese, French, German, Russian, and Spanish.

Security and Privacy. As mentioned before, security is a fundamental issue in health record systems. Therefore, during our system design and development, we profoundly considered security and privacy. First and foremost, PDF files are checked so as to detect the signature of related malicious activity. In addition, data encryption feature, safe password secure, setting password for output file, advanced logging and exception handling system are a part of this consideration.

Batch Processing. The function of batch processing is in need and is quite scare in the current tools, since the employees at many organizations are often required to deal with a great number of PDFs in the same format. For instance, an HR staff in an international corporate is asked to retrieve the name, social security number, DOB, and address of all newly employed personnel worldwide from their respective PDF paperwork, and have them all sent to another agency in a single Excel file. The ability of batch processing will allow the employee to handle this kind of task easily, saving her/him loads of frustration and time.

Ease of Use. Ease of use is the basic requirement for all tools [26]. And especially for this kind of tool, users should be given one direct way to finish the task they want. Most existing software in this area are quite complex to operate, whereas CyberPDF has a straightforward three-step workflow that can be used by any type of user from low to high computer literacy.

Free and Open-source. CyberPDF is completely open source and free of charge to use, via the provided GitHub link in the preceding section. With roots in the open source, we believe the tool would able to evolve more quickly, with bugs detected by the interested users. It would be also considered more trustworthy with source code open and available to inspect. This way researchers or developers can see for themselves exactly what it does. They can verify whether or not it is secure and introduce fixes and improvements if necessary.

5 CONCLUSION

Data processing for PDF documents is an important part of the modern enterprise operation and society, particularly in sectors where accuracy of data processing is important. One of the most important applications is to extract and transfer the data from PDF files in a batch (such as bill invoices) and migrate them to the appropriate files (Such as Excel files) or even third-party websites. Currently, most companies or businesses deal with this kind of data processing in an out-of-the-date manner, which relies heavily on manual labor, inevitably imposing a lot of errors and time delay in the process. In the market, there are also some outsourcing companies to accept the enterprise of this file processing tasks.

This research took the first step to alleviate such problems by proposing an automated solution with a tool support. It is safe to say that the proposed tool positively meets the requirements of batch processing, ease of use, and security and privacy that have been missed by similar existing tools.

In the near future, we intend to provide an empirical analysis of the tool with respect to specific features of it including, efficiency, accuracy, and security.

REFERENCES

[1] B. Yildiz, K. Kaiser, S. Miksch, pdf2table: A Method to Extract Table Information from PDF Files, Proceedings of the 2nd Indian International Conference on Artificial Intelligence, 1773-1785 (2005).
[2] N A. Rosenberg, DISTRUCT: a program for the graphical display of population structure, Molecular Ecology Resources, 4(1), 137-138 (2004)
[3] L. Leurs, The history of PDF, Retrieved on, 9-19 (2007)
[4] D. Johnson, The 8 most popular document formats on the web, Retrieved, 2, 14-30 (2014)
[5] A L. DeFranco, R S. Schmidgall, Cash Budgets, Controls, and Management in Clubs, The Journal of Hospitality Financial Management, 25(2), 112-122 (2017)
[6] C. Ramakrishnan, A. Patnia, E. Hovy, Layout-aware text extraction from full-text PDF of scientific articles, Source code for biology and medicine, 7(1), 7 (2012)
[7] D. Jiang, X. Yang, Converting PDF to HTML approach based on Text Detection, Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, 982-985 (2009)
[8] M. Aiello, C.Monz, L.Todoran, M.Worring, Document understanding for a broad class of documents, International Journal on Document Analysis and Recognition, 5(1), 1-16 (2002)
[9] E. Bart, P.Sarkar, Information extraction by finding repeated structure, Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, ACM, 175-182 (2010)
[10] KC. Santosh, A.Bela, Pattern-based approach to table extraction, Iberian Conference on Pattern Recognition and Image Analysis,Springer, 766-773 (2013)
[11] T. Hassan and R. Baumgartner, Table Recognition and Understanding from PDF Files, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 1143-1147 (2007).
[12] D. Maiorca, I. Corona, G. Giacinto, Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection, Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, 119-130 (2013)
[13] A. Lin, S. Prish, S. Der, Associating captured image data with a spreadsheet, U.S. Patent, 9,042,653 (2015)
[14] G G Parker, Alstyne Van, Two-sided network effects: A theory of information product design, Management science, 51(10), 1494-1504 (2005)
[15] B C. Lear, C E. Merrill, J M. Lin, AG protein-coupled receptor, groom-of-PDF, is required for PDF neuron action in circadian behavior, Neuron, 48(2), 221-227 (2005)
[16] Y. Aytar, A. Zisserman, Tabula rasa: Model transfer for object category detection, Proceedings of the IEEE International Conference on Computer Vision (ICCV). IEEE, 2252-2259 (2011)
[17] PDFBox A, Apache PDFBox (2017)
[18] C L R. MendonÃğa, A M R. Vincenzi, DÃŋvida TÃĺcnica: um estudo de caso com produtos de cÃşdigo aberto, thesis (2014)
[19] Apache P O I, Java A P I, To Access Microsoft Format Files, http://poi. apache.org (2009)
[20] P. Gobin, H H. Hall, C R. Hauryluck, Web based integrated customer interface for invoice reporting, U.S. Patent, 6,745,229 (2004)
[21] A. Alorf, A L. Abbott, Improved Face and Head Detection Based on Traditional Middle Eastern Clothing, International Conference Image Analysis and Recognition, 389-398 (2017)
[22] Kruse, C. S., Smith, B., Vanderlinden, H., & Nealand, A. (2017). Security Techniques for the Electronic Health Records. Journal of Medical Systems, 41(8), 127. http://doi.org/10.1007/s10916-017-0778-4

[23] S. Walker-Roberts, M. Hammoudeh and A. Dehghantanha, "A Systematic Review of the Availability and Efficacy of Countermeasures to Internal Threats in Healthcare Critical Infrastructure," in IEEE Access, vol. 6, pp. 25167-25177, 2018.

[24] W. Meng, K. K. R. Choo, S. Furnell, A. V. Vasilakos and C. W. Probst, "Towards Bayesian-Based Trust Management for Insider Attacks in Healthcare Software-Defined Networks," in IEEE Transactions on Network and Service Management, vol. 15, no. 2, pp. 761-773, 2018.

[25] Maiorca, D., Corona, I., & Giacinto, G. Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, pp. 119-130 (2013).

[26] Reza M. Parizi, Abdul Azim Ghani, Sai Peck Lee, and Saif Ur Khan. 2017. RAMBUTANS: automatic AOP-specific test generation tool. Int. J. Softw. Tools Technol. Transf. 19, 6 (November 2017), 743-761. DOI: https://doi.org/10.1007/s10009-016-0432-3

[27] Reza M. Parizi, Abdul Azim Ghani, "Architectural Knowledge Sharing (AKS) Approaches: a Survey ResearchâĂİ, Journal of Theoretical and Applied Information Technology, 1224-1235 (2008).