Filtering Email Addresses, Credit Card Numbers and searching for Bitcoin Artifacts with the Autopsy Digital Forensics Software

Patricio Domingues[1,2,3], Miguel Frade[1,3], and João Mota Parreira[4]

[1] ESTG - Polytechnic Institute of Leiria
[2] Instituto de Telecomunicações
[3] CIIC – Computer Science and Communication Research Centre
[4] Void Software, SA
Portugal

Abstract. Email addresses and credit card numbers found on digital forensic images are frequently an important asset in a forensic casework. However, the automatic harvesting of these data often yields many false positives. This paper presents the *Forensic Enhanced Analysis* (FEA) module for the Autopsy digital forensic software. FEA aims to eliminate false positives of email addresses and credit card numbers harvested by Autopsy, thus reducing the workload of the forensic examiner. FEA also harvests potential Bitcoin public addresses and private keys and validates them by looking into Bitcoin's blockchain for the transactions linked to public addresses. FEA explores the report functionality of Autopsy and allows exports in CSV, HTML and XLS formats. Experimental results over four digital forensic images show that FEA eliminates as many as 40% of email addresses and 55% of credit card numbers.

Keywords: digital forensics, email addresses, credit card numbers, Bitcoin

1 Introduction

The ubiquity and omnipresence of digital devices in our lives, either as computers, smartphones or IoT devices, has made digital forensics almost unavoidable when a abnormal situation, such as physical, financial or cyber crime, occurs. An email address is often an high value asset in a digital forensics casework. Indeed, finding email addresses in a digital device may allow to establish connections between individuals or to detect email accounts that were unknown to the examiners [15]. Also, many online services, such as social networks, rely on email addresses for authentication. All of this highlights the importance of email addresses in a digital forensic casework. When harvesting data in a digital forensic image, the digital forensic Autopsy software[1] resorts to regular expressions (*regex*) to collect strings and create a list of potential email addresses. However,

[1] `www.sleuthkit.org/autopsy/`

the regex-based list often has a high percentage of false positives, which adds noise and burden to the forensic examiner.

Credit cards have, since their inception, been the target of frauds and theft. This trend continues in the digital age, with credit cards being an important method of payment [2]. Credit cards are involved in several types of crimes. For instance, fraudulently obtained credit card numbers (spied, illegally bought, *etc.*) can be cloned to bogus cards, or simply abused for online transactions. Credit cards can also have a relevant role in an investigation by placing a given individual in a place at a given time, or having performed an action such as buying an item of interest (*e. g.* ammunition). Therefore, in some cases, it might be important in the context of a digital forensic analysis to detect credit card numbers that exist in a digital forensic image. Similarly to email addresses, Autopsy relies on *regex* to harvest credit card numbers, which yields numerous false positives that might obfuscate the forensic analysis.

Since the inception of Bitcoin in 2009, cryptocurrencies have gained significant traction and valuation. Its anonymity has attracted malicious actors to use cryptocurrencies to hide financial transactions, collect ransom from ransomware schemes [10], just to name a few. Additionally, valuation of cryptocurrencies has attracted malicious users with the intent of robbing funds, or to steal CPU and GPU cycles to mine cryptocurrencies [5]. Bitcoin was the first cryptocurrency and is still the most used and valued one. Currently, no support exists within Autopsy to collect Bitcoin-related artifacts.

We present the *Forensic Enhanced Analysis* (FEA) report module for the digital forensic software Autopsy. The main goal of FEA is to filter out false positives from the lists of email addresses and credit card numbers, as well as, presenting a list with Bitcoin public addresses and private keys, if any are found. To detect invalid addresses, FEA resorts to various techniques such as validating the employed set of characters, verifying the existence of the domain name, and checks its existence in the Internet Archive `archive.org`. The credit card numbers are verified through the validation of their check digit. Additionally, the FEA is able to harvest potential public addresses, and private keys, of Bitcoins and then to search them in the transactions ledger. Since FEA[2] is a report generator for Autopsy, all of its output is available through reports.

The main contributions of FEA are the ability to filter out *i*) invalid email addresses, *ii*) invalid credit card numbers and *iii*) to report on Bitcoin artifacts, namely public addresses and private keys found in digital forensic images. Features *i*) and *ii*) aim to reduce noise, contributing to reducing the volume of data to be analyzed by the forensic team, while *iii*) aim to harvest and filter Bitcoin related data, if any, that can be relevant for some investigations.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 briefly presents the digital forensic software Autopsy, while Section 4 describes the main characteristics of the FEA module. Section 5 describes the experimental scenario and the main experimental results of FEA/Autopsy. Finally, Section 6 concludes the paper and points out possible future work.

[2] FEA is available at `https://doi.org/10.5281/zenodo.1006703` (GPLv3 license)

2 Related Work

Validating an email address is, at first glance, simple: attempt to contact the destination email server. If the destination mail server cannot be contacted, for instance, because the domain does not exist, or no MX records pointing to email servers exist on DNS records, then the domain, and consequently the email address is invalid [4]. On the contrary, if an email server for the destination can indeed be contacted, then it suffices to ask the email server whether the email address under scrutiny exists or not. RFC 5321 [9] defines that a request for a non existent email address should trigger from the server the answer *550 "no such user"*. Nonetheless, the 550-response is not used by mail servers, since in the past it had been abused by spammers to validate email addresses and to test randomly generated addresses or common names. Instead, when receiving a mail request for a non existent email address, the mail server just acts as the address is valid, pretending to accept the email. Even if a remote email server was honoring RFC 5321 by properly answering for email addresses validation, this most certainly could not be used in the context of a digital forensic examination. Requesting the validation of a given email address could alert the targets of the investigation and/or modify their behavior. Additionally, the elapsed time between an event under investigation and the digital forensic examination can be quite long, ranging from days to months, if not years. Within this time range, a given DNS domain, which was valid when the facts under investigation occurred, might have expired and no longer be valid when the forensic analysis is performed.

Rowe et al. [15] resort to Bayesian networks to detect false positives in the lists of potential email addresses when forensically analyzing digital storage. In their tests, Rowe et al. point out that 73% of the email addresses were classified as false positives and were thus removed from the analysis. Note however, that their main goal was to detect connection on social media accounts based on the email addresses.

As reported by Garfinkel [7], several digital forensic tools resort to *regex* to search for telephone numbers, email addresses, credit card numbers and some other similar artifacts. The Autopsy software follows this approach. It also allows users to define their own *regex* to harvest the content of digital forensic images. However, results produced by *regex* include a high percentage of false positives. FEA aims precisely to remove false positives from the list of email addresses and credit card numbers returned by Autopsy. Moreover, FEA also resorts to *regex* to build a list of potential Bitcoin public addresses and private keys. FEA further filters out the list, applying validation rules proper to Bitcoin public addresses and private keys, to eliminate false positives.

3 The Autopsy Software

The Autopsy software aggregates within a graphical user interface a wide set of functionality and tools for the analysis of digital forensic images. To access the file systems, files and directories of the forensic images under analysis, Autopsy resorts to the Sleuth Kit (STK)[3]. STK can process several types of file systems and file formats used to archive disk forensic images, such as Encase file format (E01), AFF [6], raw/dd, vmdk, vhd, just to name a few. Besides STK, Autopsy relies on other software to perform some forensic actions. For instance, to recover and carve deleted data, Autopsy resorts to the external tool *PhotoRec*, while data from Windows OS registry is parsed and interpreted by the *RegRipper* tool [12]. An ingest operation consists in parsing and processing the data from the forensic disk images. Autopsy can be extended through modules, allowing for the addition of new features. These modules can be switched on/off as per user request at the start of any ingest operation.

Two of the services of Autopsy are the creation of lists holding *i*) email addresses and *ii*) number of credit cards. However, both lists present a high percentage of false positives. From the forensic point of view, it is important to filter out the lists, to reduce the workload of the forensic examiner. FEA does not intervene on the methodology used by Autopsy to harvest and create these lists, instead, FEA filters them out since both of them are made available as artifact lists. This approach can better withstand future internal changes of Autopsy that can affect the creation of the lists.

4 The FEA module

FEA is a module for Autopsy, developed in Jython and with a Java Swing configuration interface. The module follows the report paradigm, one of the three module paradigms available within Autopsy. A report module is used to organize data in a report format defined by the code of the module. FEA can produce reports for the three main types of data that it processes: email addresses, credit card numbers and Bitcoin addresses. Non valid data are labelled as false positives. The filtering techniques for each category of data are described next.

4.1 Validating Email Addresses

Checking the syntax. The syntax of an email address obeys the complex rules defined in RFC 5322 [14]. Many of the extreme cases defined in RFC 5322 are not supported by traditional email systems. For instance, RFC 5322 allows for the inclusion of a comment in the first part of the email – before the @ symbol – with the comment being delimited by round brackets. For example, `name(comment)@example.com` and `(comment)name@example.com` are, from the point of view of RFC 5322, accepted as valid, with both being interpreted as the address `name@example.com`. However, in practice, the email services only allow, for an email address, alphanumeric symbols, dot (.), underscore (_) and dash (−). To validate the syntax of email addresses, FEA follows the approach of the main email systems. For instance, there is a maximum of 64 characters for the local part of the address and a total of 254 characters for the whole address.

[3] `www.sleuthkit.org`

Validating top level domain. The top level domain (TLD) of an address corresponds to the last section of the domain. For instance, `COM` in the case of `example.com`. The top level domains are defined by IANA and include, besides the traditional `EDU`, `COM`, `NET`, `ORG`, `GOV`, `MIL` and `INT`, the domains of countries. These domains use the two-character code that corresponds to the ISO-3166 [13] standard. Lately, the number of top level domains has increased significantly [1]. FEA uses the TLD list maintained by IANA[4]. Since the list is updated every day, FEA attempts to download the updated list before processing the list of email addresses harvested by Autopsy. For every address in the list, FEA checks whether the top level domain exists on the TLD list, removing the non-existent ones. This way, email addresses whose TLD is invalid are discarded without the need to query the Domain Name System (DNS) service.

Validating domain with DNS. The domain of an email address is the section that is on the right of the @ symbol. The validation of the whole domain is done by querying the DNS service for its existence. Thus, for every potential domain, a DNS query is performed. To hide the latency of the DNS service, FEA supports multithreading, launching several queries in parallel, one per thread. The number of simultaneous DNS queries can be defined by the the forensic examiner when launching FEA. Therefore, it is up to the user to decide whether he/she wants a faster execution at the cost of an higher load on DNS, or restrict the parallelism to control the load on DNS, thereby lengthening execution times. Note that some applications such as browsers are known to be DNS-intensive [3], without impacting much the DNS service, which is a highly scalable service [8].

History analysis at *Internet Archive*. A currently nonexistent domain might have existed in the past. Since the registration of a domain is valid for a given amount of time, if the registration is not extended, nor is the domain taken by another entity, the domain ceases to exist. For the purpose of the validations performed by FEA, it would be convenient to be able to consult the history of DNS domains. However, this approach is not viable for FEA due to two main reasons: *i*) existing services do not maintain history records for all existing top level domains and *ii*) they are not free. To circumvent these limitations, FEA resorts to the *Internet Archive*[1] to assess whether a given domain existed in the past. Thus, for syntactically correct domain names whose top level domain exists, but which are currently not registered, a lookup is performed through the Internet Archive. The project has been active for more than 20 years and thus has a significant amount of data. FEA queries the Internet Archive through the online API, and if the queried domain exists within the archive, FEA returns the date of the last time the domain was indexed. To avoid overloading the service, FEA only queries the Internet Archive in sequential mode, having no more than one query pending. Note that a positive answer from the Internet Archive – the domain has existed in the web – does not mean that the domain had an associated email service. Similarly, a domain might have existed in the past, but may have never been captured by the Internet Archive.

[4] Available at `data.iana.org/TLD/tlds-alpha-by-domain.txt`

4.2 Credit Card Numbers

The credit card module of FEA aims to validate the credit card numbers harvested by Autopsy. If instructed to do so at ingest time, Autopsy builds a list of potential credit card numbers. These potential credit card numbers are available under the result set of artifacts. To filter the list of harvested credit card numbers, FEA simply verifies, for each candidate number, whether the number obeys the LUHN's checksum[5], as it is required for any valid credit card number. The LUHN checksum is a simple and public domain validation mechanism that appends a check digit at the end of the number that it aims to protect from input errors. The check digit is computed such that the validation yields a modulo 10 number, when the whole number is correct, hence its another names, "modulo 10" or simply "mod 10".

LUHN's checksum is used with credit card numbers and with International Mobile Equipment Identity (IMEI) of mobile devices, to cite just the most important domains of usage. LUHN's algorithm only protects against certain accidental errors, such as single mistyped digit, where, for example, a 2 is entered instead of a 9 [16]. Other detected errors include some, but not all, transposition of digits, namely where two digits are swapped (e.g, 12 instead of 21). Nonetheless, and despite certain errors that are not detected, the LUHN verification allows to swiftly filter out some invalid candidate credit card numbers, thus removing false positives.

4.3 Bitcoin

Cryptocurrencies have emerged in the past decade with the main goal of allowing financial transactions to be performed without the control of a central authority. While traditional transactions rely on intermediary agents such as banks or other types of financial institutions, cryptocurrencies allow for transactions to flow directly between two entities, relying on a network of peer-to-peer nodes and on public/private cryptographic keys to authenticate transactions. From the point of view of cryptocurrencies supporters, the main advantages are the *i*) anonymity that can be achieved by having non-centralized transactions and the *ii*) lower costs of performing transactions. Since transactions can be done anonymously, cryptocurrencies are increasingly linked to crime, like for instance ransomware attacks [10], cryptojacking abuses where CPU cycles are stolen to engage in cryptocurrency mining [5], to mask illicit incomes, and, more broadly, to engage in malicious activities.

Bitcoin was the first cryptocurrency, being introduced through the seminal white paper “Bitcoin: A peer-to-peer electronic cash system” authored by a mysterious Satoshi Nakamoto [11]. The paper proposed a novel distributed infrastructure – blockchain – that comprises a sequential set of numerically identified

[5] ISO/IEC 7812-1:2006. Identification cards – Identification of issuers – Part 1: Numbering system

blocks, where each block is chained to its predecessor by a cryptographically secure link, and holds the transactions accepted by the Bitcoin network. A transaction in Bitcoin requires *i*) funds, *ii*) a public address of the receiving side of the transaction and *iii*) the private key needed to authorize the transfer of funds.

Public Addresses. Public addresses can either be regular addresses or compressed ones. The former comprise 65 bytes, while compressed public address are smaller, with 33 bytes, hence the name. A public address is the public part of the private/public key pair, and it is derived from the private key. In Bitcoin, public addresses are represented for human purposes as hash string using the base58 character set. Base58 corresponds to the well known base64 sets, stripped of the visually ambiguous characters `0` (zero), `O` (capital o), `I` (capital i) and `l` (lower case L). Additionally, the slash (`/`) and the plus (`+`) characters are not part of base58. Format-wise, the hash string of a public address has a length that ranges between 26 to 35 base58 characters. A public key hash must obey the following format: the last four bytes of the public address must match the first four bytes of a double SHA256 hash computed over the first 21 bytes of the address. This way, the last four bytes act as a checksum. FEA uses this property to validate potential public Bitcoin addresses.

Private Keys. In textual representation, a private key has 50 or 51 base58 characters and starts either with a `K`, a `L`, or a `5`. Validation of a potential private key is done indirectly through the associated public key/address. For this, the Elliptic Curve Digital Signature Algorithm is used to obtain the corresponding public key. Then, this public key is validated as above, with the last 4 bytes compared to the double SHA256 of the first 21 bytes. The rationale is that if a valid public address is generated, then the Blockchain API will return a result, even if the public key has never participated in a transaction. Even though private keys obtained with this approach may lead to unused public keys, they are still relevant information, as they may provide proof of intent to engage in illicit activities within the broader scope of an investigation.

Harvesting Potential Bitcoin keys. To harvest potential Bitcoin public addresses and private keys, FEA follows Autopsy's approach with potential email addresses and credit card numbers: it resorts to the *keyword search* module, creating a custom list with Java-based *regex*. Two *regex* for harvesting Bitcoin keys are used: one for public addresses, another one for private keys. Both are shown in Table 1. Similarly to the email addresses and credit card numbers, prior to running FEA for a report on potential Bitcoin artifacts, the user needs to run the keyword search module during the ingest stage with the Bitcoin option activated. This triggers text extraction and indexing by the Apache Solr engine, which is used by Autopsy to implement keyword search. As these *regex* are not part of Autopsy, they need to be installed by the user through the manage lists option. This can be achieved via the "Options" menu in Autopsy, under the "Keyword Search" tab.

Online Search for Transactions. Besides the local validation of both public addresses and private keys, if instructed to do so, FEA can also research within Bitcoin's blockchain the existence of transaction data involving the har-

Table 1. Regular expressions to harvest Bitcoin artifacts

Public addresses:	`^[13][a-km-zA-HJ-NP-Z1-9]{25,34}$`
Private keys:	`^[5KL][1-9A-HJ-NP-Za-km-z]{50,51}$`

vested and validated Bitcoin public addresses. As all valid Bitcoin transactions are recorded in the blockchain and often Bitcoin is used to hide identities, the search might reveal valuable data.

5 Experimental Results

FEA was assessed on four digital forensic images identified as *Win7*, *Win8.1*, *Win10* and *macOS*. Each name matches the installed OS. Besides having different OS, the forensic images have different levels of activity. Specifically, Win8.1 and macOS are from systems with low activity and few installed applications, while Win7 and Win10 images have a larger number of installed applications and have seen many more hours of usage. The tests were conducted with Autopsy software 4.8, which was the latest available version at the time of the experiments.

The adopted modus operandi for processing each forensic image was as follows. First, the PhotoRec Carver ingest module of Autopsy was run to recover content, namely deleted files, from unallocated space in disk. Then, the keyword ingest module was run, with the *Email Addresses*, *Credit Card Numbers* and *Bitcoin* options activated. Finally, the FEA report module was run, first to produce the report with the filtered out email addresses and a second time for credit card numbers. The Bitcoin report was also run, but no Bitcoin artifacts were found on the images. Thus, only results for email addresses and credit card numbers are considered.

5.1 Main Results

Email addresses. Main results of FEA filtering of candidate email addresses are shown in Table 2. The columns show, respectively, the total of candidate email addresses (*Email Addresses*), addresses discarded due to bad syntax (*Non valid syntax*), addresses with non-existent TLD (*Non valid TLD*), addresses with non valid domains (*Non valid domain*). Finally, the last two columns hold the total of non valid addresses that were removed by FEA, as well as, the respective percentage.

Syntax validation has a residual contribution to filter out addresses, contributing with less than 1% of the eliminated email addresses. Surprisingly, filtering out addresses with non-existent top level domains (TLD) has extreme results across the forensic images: Win7 has nearly half of the candidate email addresses with bogus TLD, while all others have none. Domains not found on DNS have a major role in filtering out addresses, spotting an average of roughly 559 false addresses across the four images. Overall, FEA marked roughly 39.90% of the potential email addresses as invalid.

Table 2. Filtered out emails addresses by FEA

	Email addresses	**Non valid syntax**	**Non valid TLD**	**Non valid domain**	**Non valid (all)**	**non valid (%)**
Win7	1 488	20	742	158	*920*	*61.83%*
Win8.1	820	16	0	247	*263*	*32.07%*
Win10	1 853	37	0	368	*405*	*21.86%*
macOS	4 979	13	0	1 271	*1 284*	*25.79%*
Average	**2 429**	**16.33**	**247.33**	**558.67**	**822.33**	**39.90%**

Credit card numbers. Table 3 shows the results of FEA validation of credit card numbers. The number of potential credit card number is quite significant, ranging from 17 331 (Win8.1) to 59 855 (Win10). By simply applying LUHN's checksum validation, FEA discards as false positives around 55% of the potential credit card numbers. Nonetheless, due to the huge absolute number of potential credit card numbers, the remaining 45% still represent a large volume of data.

Table 3. Filtered out credit card numbers by FEA

	Credit card numbers	**Non valid (LUHN)**	**Non valid (%)**
Win7	35 477	4 271	12.04%
Win8.1	17 331	11 617	67.03%
Win10	59 855	45 866	76.63%
macOS	41 077	26 807	65.26%
Average	*38 435*	*22 140.25*	*55.24%*

6 Conclusion

FEA is a report module for the Autopsy digital forensic software. FEA eliminates some of the false positives email addresses and credit card numbers harvested by Autopsy. To detect false email addresses, FEA resorts to several validation techniques, such as syntax verification, assess the existence of top level domains, and DNS queries of email domains. For credit card numbers, FEA applies LUHN's checksum validation, removing credit card candidates that fail the verification. Additionaly, FEA also harvests digital forensic images for potential public addresses and private keys of the Bitcoin cryptocurrency. It verifies that the found artifacts not only obey Bitcoin addresses and keys rules, but also checks for the existence of transactions involving the Bitcoin addresses in the blockchain. Being a report module, FEA presents its results through text, `CSV` and `XLS` files.

When tested with four digital forensic images, FEA marked as many as 40% of candidate email addresses as false positives and slightly more than 55% of potential credit card numbers as invalid. All of this is achieved through low demanding computational techniques, with the major used resource being queries to DNS and to the Internet Archive.

Although FEA's results are interesting, since they remove stale data from a digital forensic analysis, the large volume of remnant data, meant that other techniques need to be added to filter out more false positive artifacts. FEA's ability to filter out valid but non interesting addresses can be improved by supplying *remove lists*. Examples include *contoso.**, which is a fictitious company that appears in some Microsoft software, as well as, the non usable *example.** domains. We plan to address these and other issues in future work.

Acknowledgements

This work was partially supported by FCT, Instituto de Telecomunicações under project UID/EEA/50008/2013 and CIIC under project UID/CEC/04524/2016.

References

1. Internet corporation for assigned names and numbers: An overview
2. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. Expert Systems with Applications 51, 134–142 (2016)
3. Duchamp, D., et al.: Prefetching Hyperlinks. In: USENIX Symposium on Internet Technologies and Systems. pp. 12–23 (1999)
4. Elz, R., Bush, R.: Clarifications to the DNS Specification. Tech. rep. (1997)
5. Eskandari, S., Leoutsarakos, A., Mursch, T., Clark, J.: A first look at browser-based Cryptojacking. arXiv preprint arXiv:1803.02887 (2018)
6. Garfinkel, S.: Aff and aff4: Where we are, where we are going, and why it matters to you. In: Sleuth Kit and Open Source Digital Forensics Conference (2010)
7. Garfinkel, S.L.: Digital media triage with bulk data analysis and bulk_extractor. Computers & Security 32, 56–72 (2013)
8. Jung, J., Sit, E., Balakrishnan, H., Morris, R.: Dns performance and the effectiveness of caching. IEEE/ACM Transactions on networking 10(5), 589–603 (2002)
9. Klensin, J.: RFC 5321: Simple mail transfer protocol (2008), `https://tools.ietf.org/html/rfc5321`
10. Liao, K., Zhao, Z., Doupé, A., Ahn, G.J.: Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In: Electronic Crime Research (eCrime), 2016 APWG Symposium on. pp. 1–13. IEEE (2016)
11. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
12. Panchal, E.P.: Extraction of Persistence and Volatile Forensics Evidences from Computer System. International Journal of Computer Trends and Technology (IJCTT)-volume Issue5-May (2013)
13. Postel, J.: Domain name system structure and delegation (1994)
14. Resnick, P.: RFC 5322: Internet message format (2008), `https://tools.ietf.org/html/rfc5322`
15. Rowe, N.C., Schwamm, R., McCarrin, M.R., Gera, R.: Making Sense of Email Addresses on Drives. The Journal of Digital Forensics, Security and Law: JDFSL 11(2), 153 (2016)
16. Wachira, W., Waweru, K., Nyaga, L.: Transposition error detection in luhn's algorithm. International Journal of Pure and Applied Sciences and Technology 30(1), 24 (2015)