HAL
open science

Watch Out! Doxware on The Way...

Routa Moussaileb, Charles Berrti, Guillaume Deboisdeffre, Nora Cuppens, Jean-Louis Lanet

To cite this version:

Routa Moussaileb, Charles Berrti, Guillaume Deboisdeffre, Nora Cuppens, Jean-Louis Lanet. Watch Out! Doxware on The Way.... CRiSIS 2019 - 14th International Conference on Risks and Security of Internet and Systems, Oct 2019, Hammamet, Tunisia. hal-02313650

HAL Id: hal-02313650

https://hal.science/hal-02313650v1

Submitted on 5 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Watch Out! Doxware on The Way...

Routa Moussaileb[1,2], Charles Berti[1], Guillaume Deboisdeffre[1], Nora Cuppens[1], and Jean-Louis Lanet[2]

[1] IMT Atlantique, France
routa.moussaileb@imt-atlantique.fr
charles.berti@imt-atlantique.net
guillaume.deboisdeffre@imt-atlantique.net
nora.cuppens@imt-atlantique.fr
[2] LHS-PEC, Inria, France
ruta.moussaileb@irisa.fr
jean-louis.lanet@inria.fr

Abstract. From spyware to ransomware to leakware, the world is on the verge of getting struck by a myriad of advanced attacks. Security researchers' main objective is protecting the assets that a person/company possesses. They are in a constant battle in this cyber war facing attackers' malicious intents. To compete in this arm race against security breaches, we propose an insight into plausible attacks especially Doxware (called also leakware). We present a quantification model that explores Windows file system in search of valuable data. It is based on some solutions provided in the literature for natural language processing such as term frequency-inverse document frequency (TF-IDF). The best top 15 file "contestants" will be then exfiltrated over the Internet to the attacker's server. Our approach delivers an observation of the evolution of malware throughout the last years. It enables users to prevent their sensitive information being exposed to potential risks.

Keywords: Doxware · Asset · Exfiltration · Content Analysis · TF-IDF · NLP.

1 Introduction

A Pact with the Devil is always made when a virus executes its payload on the victim's computer as Bond *et al.* state:"The arms race between propagation and defence will continue ad infinitum" [8].

Putting computer security on sounder footing, researchers seek to decrease attacks on companies and end users. Startling news are conveyed in Symantec's latest report published in 2019 [2]. Even though cryptojacking is down, but not out, targeted attacks blossomed by 78% in 2018. Cloud security and formjacking remain a concern for companies.

Cyber Security kill chain model consists of the attack's structure progressing through several phases. It begins with a reconnaissance and, once the control

over the victim's machine is acquired, the payload is executed. This payload marks the objectives of cyber criminals.

To respond to those cyber-attacks, several strategies exist. A well-known malware is ransomware, a type of software that encrypts users documents asking for a ransom in exchange of the key used for encryption. One countermeasure is the calculation of Shannon's entropy of user's files [14, 15, 22]. In fact, if they are encrypted their value fluctuates around 8. However, this is a reactive solution. Our goal is to be a step ahead of the attacker to prevent security breaches. Thus, it will give us a better understanding of the possible intrusions.

Analysts also joined the uphill battle against cyber-attacks. In fact, it is not affecting end users only, governmental concern is on the rise since it compromises the security and serenity of a country. The ultimate goal of any company is protecting its resources: the data. Data is the most valuable asset a person could acquire. Indeed, it has and is being used for many purposes by the attacker: lucrative opportunities enabling them a monetary gain. For example, blackmailing victims in displaying their private pictures to the public. Company wise, it could be selling the information gathered to a concurrent one, which will lead to millions of dollars in term of losses.

Risk evaluation is a necessity in all the cases. Companies should take into account the potential danger of disgruntled employees that can jeopardize their supreme interests. Initial leakware threat emerged in the late 2015 with Chimeras ransomware [3]. However, no evidence prove the exfiltration of any personal information. To the best of our knowledge, no previous research was made on a plausible doxware attack and its feasibility. For all the aforementioned reasons, we endeavor presenting Doxware techniques that could be used for victim assets' extortion.

Outline The paper is structured as follows. The context and language processing are presented in Section 2. The state of the art of is described in Section 3. Our proof of concept (POC) is developed in Section 4. Protection mechanisms are provided in Section 5. Finally, the conclusion is drawn in Section 6.

2 Context

2.1 From Ransomware To Doxware

Ransomware is a specific type of malware that encrypts victims' files [19]. A second type is ransom-locker that blocks the access to the desktop without encryption's process. Data's retrieval can be possible if the ransom required by the attacker is paid. Our main concern in this paper is crypto ransomware since they present a higher threat than locker ransomware.

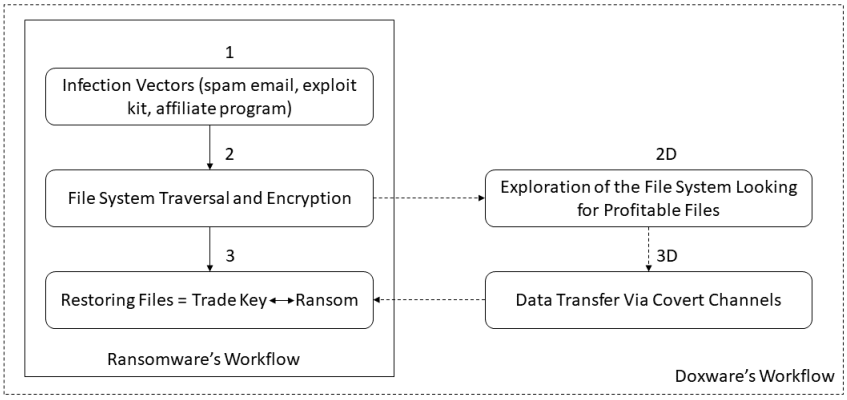

Fig. 1. Ransomware Vs Doxware.

Figure 1 presents ransomware and doxware workflow. Three main stages appear in both malware (phase 1, 2 and 3). The only difference resides in "valuable files hunting" followed by an exfiltration of the acquired data (phase 2D and 3D). To the best of our knowledge, no previous studies were made on this specific type of malware and it was only mentioned by researches as an advanced threat [10, 18, 20, 23].

Our proposed approach focuses on the file evaluation for score computation. The exfiltration phase will be presented in our future work. Doxware samples were not found on public repositories (VirusShare, MalwareDB, ...) or blogs, therefore, dynamic and static analysis were not carried out.

2.2 Data Formats Choice

Different data formats exist nowadays that are stored in a computer. They can be classified into three main categories:

1. Textual Documents: They represent files that contain mostly data in the form of a sequence of words or alphabetic characters. For example, contracts, agreements, company's balance sheet, medical records...
2. Pictures: Designs or representations made by various means (such as painting, drawing, or photography). For instance, a Magnetic Resonance Imaging (MRI), gradient descent convergence, trip pictures.
3. Videos: A recording of a motion picture or television program for playing through a television set (movies, video clip, news). These have to be personal in order to blackmail the victim in paying the ransom.

Nearly all processing methods in the literature for face recognition or body detection are based on machine learning algorithms [7, 25]. Some additional information are mandatory to be able to recognize bodies, clothes, poses...

Many drawbacks reside in these approaches such as their weight: complex algorithms requiring considerable computation power. This means many false positives cannot be tolerated. Sending 50 Mb video that does not encompass sensitive information represents a huge loss for the attacker. For example, many packets will be transmitted over the network and cannot go unnoticed. Therefore, a compromise between efficiency and stealthiness is needed. Moreover, pictures of people represent often a red line since you are affecting their privacy that means they will feel threatened. For all the reasons cited above, our proof of concept developed in this paper is based on textual documents analysis, specifically contracts.

2.3 Natural Language Processing (NLP)

- Bag-Of-Words (BOW): It is a simple technique that is part of NLP. The idea relies on regrouping words by their occurrences.
  "Mankind must put an end to war before war puts an end to mankind."
  The result will be : ("mankind" : 2),("must": 1),("put": 1),("an": 2),("end": 2),("to": 2),("war": 2),("before": 1),("puts": 1). Nevertheless, this technique has some well-known flaws. Some words, existent in any kind of documents ("the","an","is","are"...) called stop words, are not representative of the document itself compared to others. Their frequency will steal the light, as a consequence, non-relevant words will identify these documents. Hence, this technique will be backed up by the TF-IDF transformation addressing the problem encountered in BOW [26, 31].
- TF-IDF This process has the Bag Of Words as a basis but with an improved layer. To begin with, a corpus is needed, because the process compares documents one to another. The bigger the corpus, better specificities of the documents can be extracted.
  IDF : for a word i, a corpus of documents d_j (with j the index of the target document), and $|D|$ the total number of documents in the corpus, we define:

  $$idf_{i,j} = \log(\frac{|D|}{|d_j : t_i \in d_j|})$$

  For a given word i, its personal score is the logarithm of the number of documents divided by the number of documents that contain this word i. When the number of times a word is present in a document is significant, the value obtained in the logarithm is very close to 1, so idf_i will be close to 0. For a document d_j and a word, the score of this word in this particular document has a value of:

  $$tfidf_{i,j} = tf_{i,j} * idf_i$$

  having $tf_{i,j}$ the number of occurrences of the word i in the document d_j. The idf_i coefficient highlights rare words found only in few documents, even though not frequent enough, they carry some meaning and should be visible [21, 27].

– Latent Dirichlet Allocation LDA: In layman's terms, each document can be described by a distribution of topics and each topic can be described by a distribution of words. From a corpus we create the topics, so that with a single document we can link it with the appropriate topic. Considering a corpus of documents, the algorithm tries to build the topics from the content of this corpus. In the end of the execution, for each document, you get the list of the probabilities for this document to belong to the various topics [1...k]. It uses Bayesian variables to determine those probabilities. However, the number of topics are given as in input. It was not used in our current POC.

3 State of the art

3.1 Data Value

Google Scholar provides more than 5 million research papers regarding sensitive data. It is not limited to a particular field but represents a common concern for a myriad of sectors (healthcare, telecom, automotive, energy, ..). For example, mental health care is a delicate subject that could ruin a person's reputation under malicious manipulation. Netherlands data breach came mostly from the medical sector (29%) [1].

Sensitive information depends on the equipment being used. For instance, Yang *et al.* considered that the following items represent significant data on Android OS: Unique Device ID, Location, Phone number, Contact book, SMS messages and Calendar [30]. These elements carry a huge advantage. In fact, each cell phone possesses this data and any application could access it via simple API calls. Another method would be taint analysis that detects flows upcoming from known sources (IMEI of a cellphone) to untrusted sinks like the Internet. Tracking data is therefore a straightforward process in Android Devices. A similar tool developed by Sun *et al.* enables a multilevel information flow tracking by utilizing registers for taint storage, having only a 15% overhead on the CPU [24]. It presents an enhancement of TaintDroid developed by in terms of taint storage and resource consumption [9]. Considerable research is being conducted in this field as in [6, 11, 13, 28, 29].

On Android OS, attackers know what they are looking and where to find it, like extracting the GPS location of the victim. Yet, these sensitive elements cannot be predefined on a computer level. Indeed, sensitive data is only relevant to a particular end user (could be a project for a student or a painting for an artist). It exists in a variety of formats and is stored in different locations for each user.

Data's value is translated by the measure taken by a company to protect it. For instance, Zhu *et al.* provided TaintEraser a new tool that tracks sensitive user data as it flows through applications [32]. They are one of the pioneers in developing data protection from leakage on Windows OS. Their taint propagation was based on instruction and function level. They evaluated their solution

on Notepad, Yahoo!Messenger and the Internet Explorer where they presented accurate results based on taint propagation. However, TaintEraser can be bypassed via data transfer in shared memory. Loginova *et al.* suggested to use cryptographic software to carry out on-the-fly encryption [17]. They stated that it represents the most effective approach to overcome data leakage and to protect the information.

3.2 Data exfiltration

Data exfiltration is a security breach where this information is disclosed and can be published via the attacker's will. Researchers have long been interested in this domain since it can threaten a company or individual's wellbeing. Giani *et al.* revealed that the bandwidth constraints depends not only on the amount of data exchanged but also on the media being used [12]. Indeed, since 2006 little has changed. Leakage methods remain the same (FTP, SSH, email, ...).

Al-Bataineh *et al.* presented the detection of malicious data exfiltration in web traffic [5]. Their solution is based on analyzing initially the content of an HTTP POST request (using Shannon entropy) to check whether it is encrypted or not. Additional features were extracted to perform machine learning on the data gathered for malware classification.

Ahmed *et al.* tuned and trained a machine learning algorithm to detect anomalies in DNS queries [4]. Numerous elements are considered like Total count of characters in Fully Qualified Domain Name (FQDN), count of uppercase characters, count of characters in sub-domains, entropy,... Less than than 5% of false positive rate is achieved in their work. Another example is based on Liu *et al.* work, where they were able to detect data theft by analyzing the content of the data being sent to generate a signature [16]. They extracted the information from videos via wavelets enabling them to identify covert communication using Hausdorff Distance.

4 Proposed methodology

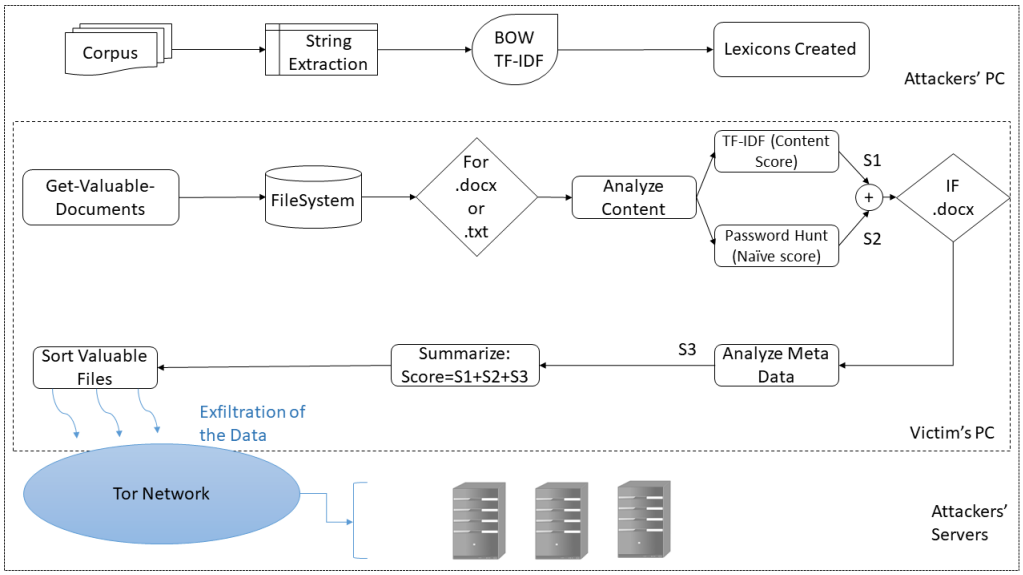

Fig. 2. Procedure's Workflow.

Procedure's Workflow is presented in figure 2. It will be thoroughly explained in the following sections.

4.1 Usable Corpus

Leakware or doxware subject is very broad and can be interpreted in many ways, whether it infects a personal or a professional machine, an individual user or a company. Our preference from among choices is the evaluation of professional documents (that can be found on both machines). Since a variety of extensions exist for textual analysis (.txt, .docx, .pdf, .rtf , .wpd, .odt...) we decided at first to restrain the study domain on .txt, .docx and .pdf files.

The content of a textual document bears its ultimate value. Therefore, the sequence of characters should be extracted for further analysis. PDF revealed later on to be quite a challenge: in fact, it could be a scanned document. Therefore, existent tools for words extraction will not be adequate. A possible solution is treating it as an image, extracting its content with the help of an Optical Character Recognition (OCR) and Tesseract (an open source OCR engine). However, many problems occurred regarding PDF files that led to discard them in our Proof of Concept. De facto, each page of the PDF document is converted into a

.png image. The ratio of PDF to image can be 1/30, a memory consuming process. The program gets importantly less stealthy. In addition to that, another noticeable problem emerged: longer processing time compared to .txt.

Concerning the Textual Documents, a usable corpus is created of various extensions (.txt, .docx) to implement the different solutions mentioned in the State of The Art. They are downloaded from *onecle.com* and *contracts-finder.service.gov.uk*. Additional noise files are gathered from Google Scholar, online courses and our own documents. This database of files can be extended in the future by adding different topics.

4.2 Evaluation Algorithm

The evaluation algorithm built depends on some parameters of the document that the program is analyzing. It is divided in two parts: one related to the attacker, as in generating intelligent lexicons in order to focus on the subjects/topics she wants to exfiltrate, and another on the victim side, evaluating the documents of the victim to send them over the network.

To accomplish these tasks four elements are required in the analysis program: Lexical Generation, Document Content Evaluation, Password files Evaluation and Meta Data Evaluation.

Lexical Generation On the attackers machine, a pre-processing is made for lexicons generation of any desired subject. To fulfill this task, a corpus of various documents based on the specific subject (.docx or .txt) is required. This topic represents the files that the attacker will be searching for on the victim's machine once infected.

Initially, TF-IDF transformation is applied on the union of documents in the corpus. The TOP -n (n an integer of your choice) results represent the words having the best score for each document, yet not all of them are equally important. For example, some acronyms like "rdc" will not be valuable on the victims pc. Since the transformation made has as an objective finding representative words for each document, then an acronym cannot be generalized to define a topic. Indeed, it is not relevant for the whole Lexicon as it must represent all documents of a specific subject.

The next step relies on creating a function that associates each word in the lexicon to an importance "score".

Let w_i be a word that represents the target subject. Let n be the number of words taken into consideration by a document (TOP n words with highest TF-IDF score). The word w_i has a $p_{i,j}$ position in the document j. Its value is built as following:

$$Sc(w_i, j) = \frac{n - p_{i,j}}{n}$$

As a result, the total score Sci will be :

$$Sc_i = \sum_{j=1}^{n} Sc(w_i, j) = \sum_{j=1}^{n} \frac{n - p_{i,j}}{n}$$

Sc_i is divided by n for normalization purposes so that any word can have a maximum score of 1. In the example of lexicon that will follow, the word "1look" arose and represents an important element in the subject corpus we acquired. Indeed, it has an important score for its own document but globally it is not representative of contract documents. A part of a lexicon produced and used on the victims machine is presented below: (1look,0.090) (company,0.6) (agreement,0.690) (section,0.272) (purchaser,0.181) (shares,0.199) (shall,0.736) (closing,0.090) (material,0.018) (date,0.009) (buyer,0.181) (acquisition,0.045).

Document Content Evaluation On the victim's side, the lexicons are already hidden in the malware source code and they will used to process a content score which will be merged with a meta data score for a final evaluation score. A modified bag of words is implemented. At first, a dictionary containing every word of the document with its number of occurrences is extracted. The initial value of the content score is 0.
Let CS be this content score, L_i the score of the word i of the lexicon being studied, n the number of words in the lexicon and occ_i the number of occurrences of the word i in the document analyzed.

$$CS = \sum_{i=1}^{n} L_i * occ_i$$

Password Evaluation Two methods are used for password evaluation:

1. A comparison of the words existing in a document is made with the 25 most common passwords (gathered from to Symantec). For each occurrence, the naive score is increased by one.
2. Hunting for password common patterns: for instance, if a word contains more than 8 characters including uppercase, lowercase, numbers, special characters and so on.
   - length : length of the word, caped between 6 and 16.
   - presence of more than two uppercase and more than one lowercase: +3.
   - presence of uppercase, lowercase and number: +3.
   - presence of a special character: +8.
   - presence of a known business (such as "facebook", "netix" and so on): +5

   Then we sum them all before dividing by 10.

Meta Data Evaluation Fifteen meta data can be accessed and extracted from a Word Document. Those are: author, category, comments, content status, created, identifier, keywords, language, last modified by, last printed, modified, revision, subject, title and version. However, most of the meta data are not filled in (the content is null), therefore, the only ones kept are the most relevant which are the number of revision, created, modified, last printed.

The algorithm Valuable File Hunting (VFHA) summarizes the steps developed in our paper.

4.3 Solution's Design

1. Initially: Lexicon Generation of Contract Topic (line 2 in VFHA).
2. Then: parsing the target file system looking for .txt and .docx extensions (line 18 in VFHA).
3. File Score
   (a) .txt: Its content is extracted and vocabulary analyzed. Each word is compared with a lexicon previously created that contains recurrent and relevant words in a contract based document. An additional comparison is made to spot if there is any noun that appeared similar to a construction of a password. Either a common one ("passwd" for instance) or since it includes special characters, uppercase, lowercase and numbers in a string. For every method called, a score between 0 and 5 is returned. In the end, those scores are summed in order to have a total that represents the value of the file taken into consideration. If the score is not null, it means that the document may yield value. Therefore, we add the file to a dictionary of potentially valuable ones linked to its score (line 3 in VFHA)..
   (b) .docx: The same procedure is done for the .docx files. However, an additional step is made for the meta data analysis. The significant metadata are the number of revision, creation date, modification date and the date when it was last printed. They will be added to the total sum representing the value of a document (line 3 and 9 in VFHA).
4. "Summarize" step: Each document has a total score that has been assigned, so the list of tuple (path, score) is sorted according to the value obtained, where the attacker chooses which ones he/she wants to extract. For instance the first 50 files (line 23 and 24 in VFHA).

Algorithm Valuable File Hunting VFH

```
procedure ALGORITHM 1
    Topic_Lexicon ← { lexicon_generator (Corpus of same Topic: Contract) }
    def analyse_content(File f):
    for word ∈ f do
        if word ∈ Topic_Lexicon then
            f_score+ = score(word) * number_occurrences
    return f_score/len(f)

    def analyse_metadata(File f):
    if f.core_properties.revision > 1 then
        f_metadata_score+ = 5
    else if f.created == "2019" then
        f_metadata_score+ = 1
    else if ... then
        ...
    return Sum(f_metadata_score)

     Parse the File System
    if FileExtension ∈ .txt or .docx then
        FileList ← {Analyse MetaData and Content}
    else
        Continue;
    Sort FileList by highest_Score
    Send n first valuable files to the attacker's server (future work)
```

4.4 Lexicon Generation

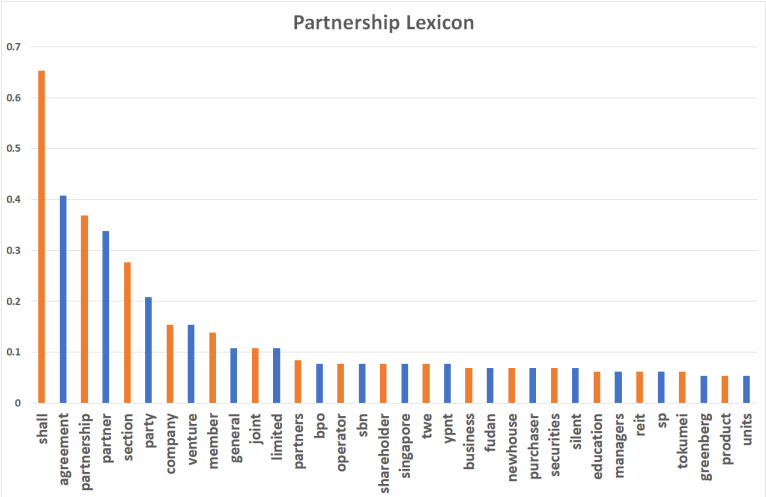

Fig. 3. An Example of Lexicon with the associated Scores.

The chart presented in figure 3 shows the most common words in a contract document are: "shall", "partnership", "agreement" and also "section". Indeed, the corpus gathered is previously identified as a contract type document, which is an advantage allowing us to perform relatively simple algorithm to determine whether a document belongs to this category or not. Although, law documents acquire also an important score and may be also as valuable as a contract.

4.5 Valuable Files Chase

Personal Computer Two versions of the algorithm are tested. The first one, heavier program since it uses scikit learn library, outputs the valuable documents in 126 seconds. During the execution, 3578 readable documents were found that had a non null score.

The second version is lighter because the scikit learn library is substituted by functions we created. Since it is not optimized, 242 seconds are required to perform a full disk analysis. Yet, the program is lighter than the initial version.

Virtual Machine A Windows 7 Virtual Machine is created for the proof of concept. It holds 50 noise documents and 25 important files. After running the algorithm, 15 files are recovered. Among those there were six false positives, and 5 were Windows configuration files. To gain better results, specific Windows file system path can be removed from the file system traversal, in addition to Read-Me related documents. The time the victim suspects something the exfiltration

process would be to an end, and the list of documents are being or sent to the attacker.

5 Security Recommendations

Protection against malware attacks, specially zero days, is a challenge for all researchers. Residual risk remains: de facto, despite various countermeasures employed by a party, an attacker can always find a way to penetrate the system (he/she still risks to be detected). If committed, anyone can reach their malicious intents. However, our goal is to complicate the intrusion task, detect it if possible, rather than handling it to the attackers on a golden plate. Users should know the existent vulnerabilities to see what patches can be used to circumvent malevolent attacks. Some countermeasure can be deployed by users to protect their data from being exfiltrated:

1. Honeypot Folders: They can be created in any environment, regardless the operating system used. Since doxware will traverse the whole file system looking for assets, any process or thread that will pass through this lure folder can be immediately flagged then stopped. A drawback would be malware's multi-threading techniques, it can still be exposed but after a certain epsilon time.
2. Data Tainting: Sensitive data in a computer is extremely private and depends on the end users, unlike Android OS (IMEI, GPS location,... existent on all mobiles). Therefore, a general protection model is impossible to develop in real life. Yet, each individual can add a layer, a taint, on his preferred/sensitive information. Thus, each exfiltration attempt over the network will be detected. Nonetheless, a person can have an explosion of tainted data that may slow down the system.
3. Data Encryption: It remains a robust way adopted by the global community. Indeed, brute-forcing the encryption key can take decades. Even though an attacker acquired the encrypted files, he/she cannot menace the victims or blackmail them since no access to the decrypted data is possible.

6 Conclusion

We have discussed in this paper the potential danger of sensitive data localization and quantification that can be carried out by a Doxware malware. Windows OS is the target system throughout the experiments. A proof of concept is developed based on contract topic and passwords hunt. To accomplish this tasks, state of the art methods were used such as TF-IDF and Bag of Words in addition to a document's meta data. The associated score of each document is calculated then normalized. To identify new target topics, few samples of files regarding the same topic are needed. Even if the victim finds out he/she got hacked, the person will not have the means to reach the attacker or react to this intrusion. New options can be added as building bricks such as PDF and Images analysis

which will strengthen the offensive invasion in attacker's point of view. Reducing false positive rate can be done by eliminating Windows system path and choosing randomly N last visited files in Windows' Quick Access. 40% of important files can be collected, by relying on a straightforward mechanism, and ready for exfiltration. Threats arising from this cyberwarfare are exponential. Therefore, end users should be aware of the possible attacks especially attack vectors in order to avoid and circumvent them protecting their assets.

References

1. Data Protection and Privacy across sectors and borders . https://bit.ly/2D2r77M
2. Internet Security Threat Report by Symantec. https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-executive-summary-en.pdf
3. New ransomware program threatens to publish user files. https://www.computerworld.com/article/3002120/new-ransomware-program-threatens-to-publish-user-files.html
4. Ahmed, J., Gharakheili, H.H., Russell, C., Sivaraman, V.: Real-time detection of dns exfiltration and tunneling from enterprise networks. Proceedings of IFIP/IEEE IM, Washington DC, USA (2019)
5. Al-Bataineh, A., White, G.: Analysis and detection of malicious data exfiltration in web traffic. In: 2012 7th International Conference on Malicious and Unwanted Software. pp. 26–31. IEEE (2012)
6. Arzt, S., Rasthofer, S., Fritz, C., Bodden, E., Bartel, A., Klein, J., Le Traon, Y., Octeau, D., McDaniel, P.: Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. Acm Sigplan Notices **49**(6), 259–269 (2014)
7. Bartlett, M.S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., Movellan, J.: Recognizing facial expression: machine learning and application to spontaneous behavior. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 2, pp. 568–573. IEEE (2005)
8. Bond, M., Danezis, G.: A pact with the devil. In: Proceedings of the 2006 workshop on New security paradigms. pp. 77–82. ACM (2006)
9. Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.G., Cox, L.P., Jung, J., McDaniel, P., Sheth, A.N.: Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS) **32**(2), 5 (2014)
10. Ensey, C.: Ransomware has evolved, and its name is doxware. DARKReading. nformationWeek Business Technology Network (2017)
11. Feng, Y., Anand, S., Dillig, I., Aiken, A.: Apposcopy: Semantics-based detection of android malware through static analysis. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. pp. 576–587. ACM (2014)
12. Giani, A., Berk, V.H., Cybenko, G.V.: Data exfiltration and covert channels. In: Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense V. vol. 6201, p. 620103. International Society for Optics and Photonics (2006)

13. Graa, M., Cuppens-Boulahia, N., Cuppens, F., Lanet, J.L., Moussaileb, R.: Detection of side channel attacks based on data tainting in android systems. In: IFIP International Conference on ICT Systems Security and Privacy Protection. pp. 205–218. Springer (2017)
14. Kharaz, A., Arshad, S., Mulliner, C., Robertson, W., Kirda, E.: {UNVEIL}: A large-scale, automated approach to detecting ransomware. In: 25th {USENIX} Security Symposium ({USENIX} Security 16). pp. 757–772 (2016)
15. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E.: Cutting the gordian knot: A look under the hood of ransomware attacks. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 3–24. Springer (2015)
16. Liu, Y., Corbett, C., Chiang, K., Archibald, R., Mukherjee, B., Ghosal, D.: Detecting sensitive data exfiltration by an insider attack. In: Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research: Developing Strategies to Meet the Cyber Security and Information Intelligence Challenges Ahead. pp. 16:1–16:3. CSIIRW '08, ACM, New York, NY, USA (2008). https://doi.org/10.1145/1413140.1413159, http://doi.acm.org/10.1145/1413140.1413159
17. Loginova, N., Trofimenko, E., Zadereyko, O., Chanyshev, R.: Program-technical aspects of encryption protection of users' data. In: 2016 13th international conference on modern problems of radio engineering, telecommunications and computer science (TCSET). pp. 443–445. IEEE (2016)
18. Lueders, S., et al.: Computer security: Enter the next level: Doxware (2017)
19. Luo, X., Liao, Q.: Awareness education as the key to ransomware prevention. Information Systems Security **16**(4), 195–202 (2007)
20. Nadir, I., Bakhshi, T.: Contemporary cybercrime: A taxonomy of ransomware threats & mitigation techniques. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). pp. 1–7. IEEE (2018)
21. Ramos, J., et al.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning. vol. 242, pp. 133–142. Piscataway, NJ (2003)
22. Scaife, N., Carter, H., Traynor, P., Butler, K.R.: Cryptolock (and drop it): stopping ransomware attacks on user data. In: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). pp. 303–312. IEEE (2016)
23. Sherer, J.A., McLellan, M.L., Fedeles, E.R., Sterling, N.L.: Ransonware-practical and legal considerations for confronting the new economic engine of the dark web. Rich. JL & Tech. **23**, 1 (2016)
24. Sun, M., Wei, T., Lui, J.: Taintart: A practical multi-level information-flow tracking system for android runtime. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 331–342. ACM (2016)
25. Sun, X., Wu, P., Hoi, S.C.: Face detection using deep learning: An improved faster rcnn approach. Neurocomputing **299**, 42–50 (2018)
26. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd international conference on Machine learning. pp. 977–984. ACM (2006)
27. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting tf-idf term weights as making relevance decisions. ACM Transactions on Information Systems (TOIS) **26**(3), 13 (2008)
28. Xue, L., Qian, C., Zhou, H., Luo, X., Zhou, Y., Shao, Y., Chan, A.T.: Ndroid: Toward tracking information flows across multiple android contexts. IEEE Transactions on Information Forensics and Security **14**(3), 814–828 (2019)

29. Yan, L.K., Yin, H.: Droidscope: Seamlessly reconstructing the {OS} and dalvik semantic views for dynamic android malware analysis. In: Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12). pp. 569–584 (2012)
30. Yang, Z., Yang, M.: Leakminer: Detect information leakage on android with static taint analysis. In: 2012 Third World Congress on Software Engineering. pp. 101–104. IEEE (2012)
31. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics **1**(1-4), 43–52 (2010)
32. Zhu, D.Y., Jung, J., Song, D., Kohno, T., Wetherall, D.: Tainteraser: Protecting sensitive data leaks using application-level taint tracking. ACM SIGOPS Operating Systems Review **45**(1), 142–154 (2011)