Monitoring Leaked Confidential Data

Slim Trabelsi
Global Security Research
SAP
Mougins, France
slim.trabelsi@sap.com

***Abstract*— During the first half of 2018 over than 945 data breaches resulted in 4.5 Billion data records been compromised worldwide. Data leak is one of the biggest security issues targeting the industrial and governmental sectors. The data loss hemorrhage is too important and uncontrollable that companies and institutions need to react very quickly to reduce the risk of being targeted by an attack exploiting leaked data. Unfortunately, this in not yet the case, because on average a company spend 196 days to identify a data breach and 69 additional days to contain it. In order to reduce the identifications time, we propose a solution to monitor, in real time, huge streams of leaked data published on hacking sources. These ese data are classified, and confidential information is precisely identified. This classification is per-formed by the combination of inference rules and a Convolutional Neural Network pre-trained model, which recognizes different patterns of confidential data. We also describe our observations from the data that we collected and identified in the context of a company monitoring use case.**

Keywords*— *Data leak, breach, password, credentials, hacking, artificial intelligence, classification, cyber security.

I. Introduction

Data breaches is one of the biggest security issues in the modern cyber world. The number of leaked records is exponentially growing every year (see Figure 1), the volume of exposed records become tremendous and hemorrhage seems to be un-stoppable. According to the Breach Level Index[1], that is a global data base measuring the public data breaches, over than 945 data breaches led to 4.5 Billion data records been compromised worldwide, only during the first half of 2018. The volume increased by 133% compared to the same period in 2017, that was already a record.

Still according to the Breach Level Index in 2018 18.525.816 records are stolen or lost every day, so around 772 per hour and 214 per second. These records contain medical, credit card and/or financial data, personally identifiable information (PII), various passwords and credentials, personal conversations, e-mails, classified documents, etc. The healthcare sector remains the most exposed to breaches in 2018 by 27% of the total number of incidents. In terms of volume of data leaked, social network platforms is leading in 2018, with 76% of the total volume of data breached, and this is mostly due to the Cambridge Analytica-Facebook incident[2], accounted for over 56 % of total records compromised.

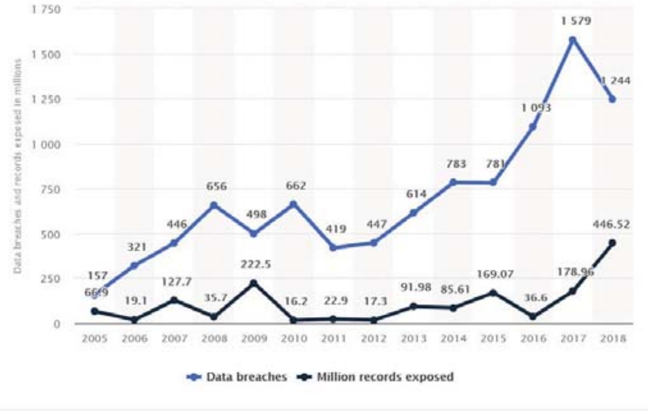

Fig. 1. Annual number of data breaches in the US from 2005 to 2018[3]

All these threatening statistics show that almost 80% of the citizens living in the industrialized countries are concerned by a data leak (personal or not). Similarly, the consequences on the personal and business level are significant. A recent study from IBM [9], published in July 2018, evaluated the global cost of data breaches[4] in the world. This report showed that the average total cost of one data breach is estimated to $3.86 million and the average cost per lost or stolen record is $148. This cost has an increase of 6.4% compared to 2017. A data breach can, in 40% of the cases, open the way for a new attack. The leaked data especially PIIs and credentials are exploited to target new victim or replay the attack for an already ex-posed victim, especially when these victims are not aware about their data breach. The larger the data breach, the less likely the organization will have another breach in the next 24 months. This is due to the investments on security after a huge data breach.

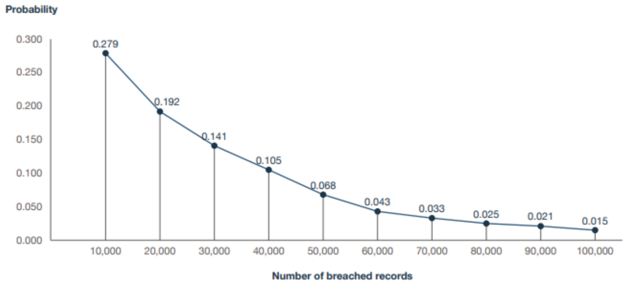

Fig. 2. Probability of a data breach involving 10000 to 100000 records [9]

Another parameter that influences directly the cost and the impact of a data breach is the time to identify and contain the

[1] https://breachlevelindex.com/

[2] https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election

[3] https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/

[4] https://databreachcalculator.mybluemix.net/assets/2018_Global_Cost_of_a_Data_Breach_Report.pdf

978-1-7281-1542-9/19/$31.00 ©2019 IEEE

breach. On average a company needs 196.7 days to identify a data breach and 69 additional days to contain it.

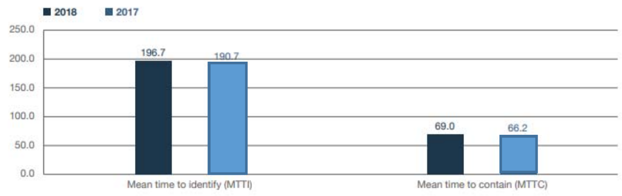

Fig. 3. Time period to identify and contain the data over the past years [9]

These delays are obviously too long to reduce the financial and image impact of a data breach. The reason is simple, there is no mean for a company to know that it was breached. Even if some services like https://haveibeenpwned.com/ exists to notify a person of a company when their credentials are leaked, it is not sufficient to cover all the types of data breaches. The response time is very critical to reduce the exposure period and the probability of multiple attacks exploiting these leaked records.

To reduce the time to identify a data breach, we propose here an automated tool that crawls several sources including internet and dark web websites containing leaked data shares. Once all the data are collected, the tool classifies the organization related data contained in these records. This classification is done using a convolutional neural network pre-trained model that recognizes different patterns of confidential data.

We tested this tool for 12 months and made statistics on what we collected in the context of the SAP[5] data leak monitoring use case.

This paper is organized as follows: in section 2 we explain the root causes of data breaches, in section 3 we detail our solution, in section 4 we introduce the SAP use case and we discuss the study observations, in section 5 we give information about the legal background of the study for GDPR compliance, in section 6 we compare our solution to the existing related work. In section 7 we conclude.

II. Root Causes

Knowing the root cause of data breaches is the first step of the prevention process. Although if all the studies and analysis converge in the causes, they almost all diverge on the distribution and the ratio of each root cause. This divergence is maybe due to the definition of each root cause and the interconnection of each cause. If we use the classification of the last IBM's report [9] we can classify the roots of data breaches in three categories defined below.

A. Malicious or Criminal Attack

This is related to all the attacks coming from malicious activities from cyber criminals that exploits system vulnerabilities to breach them using specific exploit kits or malwares introduced inside the organization. The criminal activity incudes also the social engineering attacks, that exploit the human vulnerabilities to obtain access to the sensitive data like credentials. Insider's threats can also be classified under the category criminal attack even if it comes from the employee itself. The insider attacker has three main motivations: the financial appetite, the ideological or political conviction and the revenge instinct. The physical theft falls also under the umbrella of the criminal attack.

B. Human Error

The human error is essentially the main root cause of data breach, we can see its implication in the two other causes. This human error is a mistake committed by an employee (internal non-intentional) that will open the floor for a data breach. For example, sharing a confidential record in a public repository. Misconfiguring an access control system and open the access to sensitive resources. Using easy to guess passwords, writing passwords on post-it, reusing the same password every-where, etc.

Being a victim of a simple social engineering attack could be assimilated to a mistake. Not patching and updating systems is also a human mistake that can lead to an external attack. For this reason, the human error is the central weakness at the origin of most of the external and internal attacks.

C. System Glitch

The system glitch is defined as a failure in the normal process execution of a system. In this case the basic security guards can be disabled, opening the floor to a non-authorized access to system data. Recently a system glitch in Singapore Air-lines[6] systems exposed personal data of their frequent flyer members in January 2019. This glitch provoked a bug in the authentication system that was disabled, allowing then the access to the protected data.

When a system is breached, depending on the root cause, at some point of time, the leaked records will be sold then shared for free on internet and can be exploit-ed by any malicious user or organization accessing it. And the key point to reduce this risk for a company, is to know in the early phase that its confidential assets were leaked and react very quickly.

III. Solution Design and Architecture

A. Data Gathering

In this paper we propose a solution to identify the data breach as soon as it is publicly disclosed. We do not prevent the leak before it happens, but we try to reduce the identification and containment time then reduce the exploitation exposure risk. We made a search to identify different sources on internet and on the dark web where people are used to share leaked and stolen data. Some studies were published [1] and [6] to identify these sources and observed the behavior of the populations that exploits these leaked data. We mainly focus on anonymous file and text sharing platforms like pastebin.com. In addition to this, we also focus on certain data leak exchange forums and very recently public source code repositories like Github.

For each of these sources we created crawlers to collect all the published data. Some of the platforms like Pastebin or Github offer scraping APIs that we use, and some other websites are not providing such interfaces (mainly on the dark web), so we use web crawlers. The data is collected in a streaming mode and then passed to the processing pipeline with some additional metadata related to the source and the time.

[5] www.sap.com

[6] https://www.computerweekly.com/news/252455339/Singapore-Airlines-software-glitch-exposed-customer-data

B. Data Processing

Once the data collected we apply several techniques for data identification, classification and extraction.

1) Regexp and Inference rules

In order to perform a first filtering, we use simple regular expressions to identify key items in the collected records like email addresses, IP addresses, specific key-words. This first pass is used to start classifying and tagging documents, but also extract specific items from the records to use it afterword. For example, if the document contains only e-mail addresses, it can be classified as mailing list (or spam list). If a document contains certain keywords related to a company or organization, it will be tagged as related to this company.

Inference rules comes after to introduce a certain logic with the keywords identified using regular expressions. These inference rules are used to identify a specific combination of keywords that are related to known document structures and standards. For example, if a document contains the sequence like: BEGIN PGP PRIVATE KEY BLOCK+ * + END PGP PRIVATE KEY the inference rule will identify the element as a private key of a PGP certificate.

The execution of regular expressions and inference rules gives already a first classification structure and a first identification set of the items contained in the collected documents. The limit remains in the size and the scalability of a rule-based system for an unstructured and completely heterogeneous dataset. We do not know in advance what we will collect and what will be the format. For this reason, we decided to add an additional classification layer where artificial intelligence is involved.

2) Deep learning-based classification

In order to capture the occurrence of certain keywords and their different combi-nation in a document, we use a deep learning model (namely a Convolutional Neural Network, CNN) for Natural Language Processing (NLP) at the word level (and not on the character level). This approach is used identify patterns of keyword occurrences in a non-structured document. The idea is to replace a non-exhaustive set of inference rules used to identify certain type of elements within a document, by a deep learning model predicting the occurrence elements in the same document. The first application case is the identification of login password combo files. Most of the time, such kind data is published with very few contextual information about the origin of the leak. In these cases, the released records contain only a long list of lines with this format:

$$\textit{Login} + \textit{Separator} + \textit{Password}$$

To train the CNN model, we collected 5296 records from Pastebin that we tagged manually. Then we tokenized the text by unifying character sets and new lines encoding, removal of {*,-,=} characters, identification of relevant separators {,,.,:,;,|,@, ,\t, }, removal of characters less than two letters, lowercase all the characters, and padding.

The neural network model was configured according to these following layers:

- Word Embeddings (100)
- Dropout (0.2)v1D(64, 5, relu)
- MaxPooling1D(32)
- LSTM(100, backwards)
- Dense(1, sigmoid)

The final accuracy value was 96%.

3) Scoring

During the classification process every record is tagged according to the different items that it contains. A record can have multiple tags. For example, if the record is a Java source code containing a login and password, and IP address, a person name and a company brand, five tags will be affected. Depending on the use cases, the combination of tags can influence the relevance and the sensitivity of the leaked record. For example, a Java source code with a company brand and an authentication combo is much relevant and sensitive than the same java source code without the combo. The first one then will be treated with a higher priority.

For each tag we affect a certain weight depending on the relevance and criticality. When a record is tagged with several tags, we sum the weight to have a score. This score represents a severity score used to evaluate the priority to handle this record. The scoring and the weight attribution are purely subjective and strongly depend on the use case.

C. Architecture

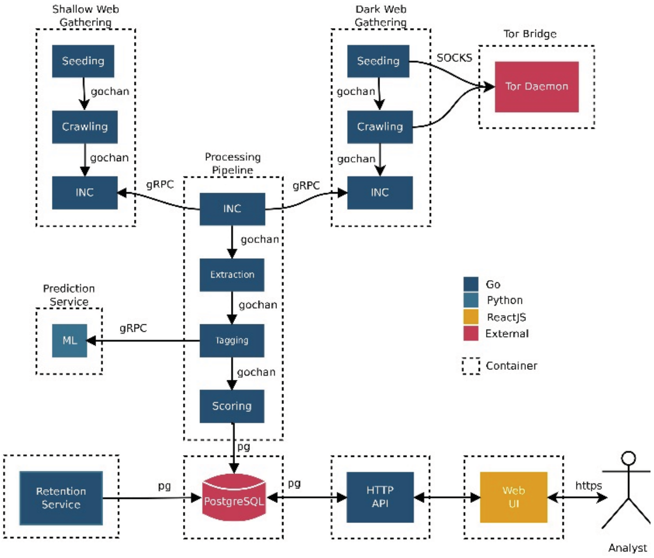

Fig. 4. Architecture of the monitoring tool

Figure 4 represents the global architecture of the tool that is deployed as a micro-services architecture where every dashed box element is running on a separate container. We can classify these elements into four categories: Data Gathering, Data Processing, Data storage and Interface.

1) Data Gathering Pipeline

Due to the additional technical constraints introduced by the TOR network access requirements, we split the data gathering module in two components: one dedicated to the dark web crawling, that is coupled with TOR bridge, and another one dedicated to the internet crawling. The data gathering pipeline has two steps: seeding that downloads a list of new records from the sources and the Crawling that downloads individual records from sources. The collected records are passed to the processing pipeline.

2) Processing Pipeline

The collected data is extracted then tagged using regular expressions and inference rules, then sent to the Prediction service where the CNN model is predicting the content of the record and add a new tag on the record. Several models can be used in parallel to predict on the content of the record. Once record is classified and completely tagged, a relevance score is computed according to the tags.

3) Storage Pipeline

All the records, the metadata, the tags, the predictions and the scores are persisted in a PostgreSQL DB. Attached to this DB we added a GDPR compliance component called Retention service that implements several methods to handle properly the collected data (containing personal information in some cases). The GDPR compliance topic will be further detailed in the paper.

4) Interfaces

In order to interact with other applications and systems, we put in place a JSON API that can be consumed by ticketing systems of security incident management platforms and automate the investigation and response process. A UI (see Figure 5) is also available for a direct user interaction with an operational dashboard.

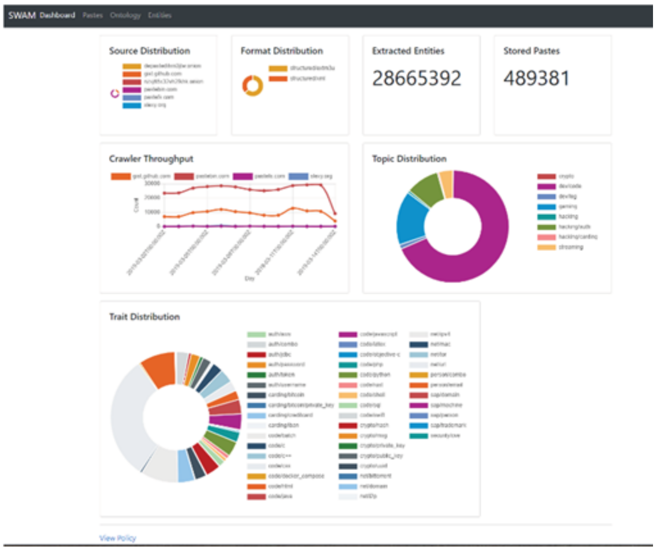

Fig. 5. Monitoring Dashboard

IV. Use Case

A. Context

In order to test the proposed solution, we decided to run it for one month to monitor the leaked data related to SAP (some details will be omitted for confidentiality). For this use case we decided to focus on the following topic of interests: Crypto data, Source Code, System Logs, Gaming, Hacking activities, Credentials, Carding, Streaming. Under these topics we identify 45 types of items including SAP internal domains, SAP trademark representation, and SAP products. We connected six sources to the input. Two sources from the dark web and four from internet including the Github public commits.

B. Observations

With the configuration described before we obtain a daily average throughput of 36.000 collected distinct records (excluding redundant posts), with more than one million records in a month. More than 2 million entities are extracted every day. Figure 6 shows the monthly global distribution of records per topic (independently from the SAP related data presence).

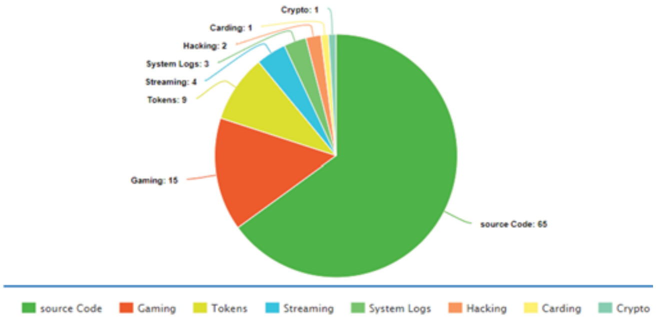

Fig. 6. Topic distribution over one month (numbers indicate percentages)

Concerning the SAP related findings, the scoring system combined with the real-time alerting and response process reduced drastically the reaction time and the company exposure risk. For example, if a password related to an SAP email address is leaked (not necessarily a corporate password), the tool will identify the leak immediately and an alert is automatically triggered, for further analysis. We also observe that diverse and unexpected types of confidential documents and information can be collected. The scope of this data is not only limited to credentials, but it also concerns internal documents, e-mails, source code, system logs, configuration files, internal reports, etc.

V. Legal Compliance

When we collect such sensitive data, the legal constraints are very strict especially with regards to the GDPR European regulation. As shown in the architecture chapter, there is a dedicated component, called retention service, in charge of enforcing the privacy rules mandatory for the compliance. This service offers a retention period for the data lifetime, and all the non-alert-related data is automatically deleted. Every personal information stored in the system can be retrieved immediately on owner's request and can be deleted if necessary.

VI. Related Work

In the last two years, many IT companies started to offer services to track leaked credentials, one of the most famous service is provided by https://haveibeenpwned.com . The concept consists in collecting manually recent data leaks then match the login and user name of the requester in order to check if his credentials are leaked. More recently similar services from start-ups like https://leak.watch diversified their offer by extending the monitoring to company or personal data. Most of these solutions are on their early phase and rely on a lot of manual search and processing. We also compared their findings to ours, and we noticed a certain gap in the timing and in the content especially for corporate confidential data like leaked source code. Most of the security big players like www.tenable.com offers a network-based asset monitoring tool that prevents data leakage from the company. This kind of approach is clearly not efficient due to the variety of leak channels and the type of leaked data. In the academic and research, most of the approaches are based on the data tracking inside the company to prevent any leak. We can site for example Papadimitriou et al. [2] that proposed to watermark the confidential and fake data within the company to trackback the origin of the leak. Schutte et al proposed to use the tainting approach to track leaked data from Android mobile phones. Cheng et al. [4] proposed to use context and content-based classification techniques in order to monitor the sensitive asset mobility over the company network. These approaches are clearly not sufficient to cover the huge amount of diverse confidential data that can be leaked through several non-controlled channels. Hauer [5] demonstrated lack of efficiency of content-aware data leakage prevention solution (DLP) confirming our security evaluation. Liu et al. [3] proposed a best practice list to reduce the risk of data leak, but they did not provide any automated solution. The idea of reporting password leak to concerned people was proposed by Malderle et al [8] but their approach is based on a manual data collection and classification.

VII. Conclusions

In this paper we propose a monitoring and alerting solution to identify when confidential data is breached and becomes publicly available. The goal of the solution is to alert as early as possible when a confidential data is leaked to reduce the exposure time and speed-up the response time. We pro-pose to crawl several hacking sources on internet and on the dark web and collect all the published information. We use an innovative hybrid approach to classify and identify the relevant confidential data. This approach is based on the combination of inference rules and deep learning prediction models. We conducted a use case study for SAP-related information, and we collected leaked confidential assets for one month. We observed that the collected data is not only composed by credentials, but various type of data is concerned, especially source code, system logs and other kind if internal documents. We also clearly perceived the impact of such approach in the response time to fix the issue and the reduction of the threat exposure period. Being able (when possible) to delete data from the leak website and identify quickly the root cause of the leak, give to organizations a significant advantage to protect the intellectual property and the private and personal data of their employees and customers.

As future work we decided to give a stronger focus on the open source code repositories like Github that, from a preliminary analysis, seems to contain more confidential data than expected.

References

[1] Missaoui, C., Bachouch, S., Abdelkader, I., & Trabelsi, S., Who Is Reusing Stolen Passwords? An Empirical Study on Stolen Passwords and Countermeasures. In International Symposium on Cyberspace Safety and Security (pp. 3-17). Springer, Cham. (2018, October)

[2] Papadimitriou, P., & Garcia-Molina, H., Data leakage detection. IEEE Transactions on knowledge and data engineering, 23(1), 51-63. (2011)

[3] Liu, S., & Kuhn, R., Data loss prevention. IT professional, 12(2), 10-13. (2010).

[4] Cheng, L., Liu, F., & Yao, D. D., Enterprise data breach: causes, challenges, prevention, and future directions. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(5). (2017).

[5] Hauer, B., Data and information leakage prevention within the scope of infor-mation security. IEEE Access, 3, 2554-2565. (2015).

[6] Jaeger, D., Graupner, H., Sapegin, A., Cheng, F., & Meinel, C., Gathering and analyzing identity leaks for security awareness. In International Conference on Passwords (pp. 102-115). Springer, Cham. (2014,).

[7] Schütte, J., Titze, D., & De Fuentes, J. M., Appcaulk: Data leak prevention by injecting targeted taint tracking into android apps. In 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (pp. 370-379). IEEE. (2014).

[8] Malderle, T., Wübbeling, M., Knauer, S., Sykosch, A., & Meier, M., Gathering and analyzing identity leaks for a proactive warning of affected users. In Proceedings of the 15th ACM International Conference on Computing Frontiers (pp. 208-211). ACM. (2018).

[9] Ponemeon Institute. 2018 Cost of a Data Breach Study: Global Overview. July 2018.