Track Down Identity Leaks using Threat Intelligence

Timo Malderle[1], Sven Knauer[1,2], Martin Lang[3], Matthias Wbbeling[1,2] and Michael Meier[1,2]
[1]*University of Bonn, Germany*
[2]*Fraunhofer FKIE, Bonn, Germany*
[3]*Independent Researcher, Leverkusen, Germany*

Keywords: Identity Leakage, Treat Intelligence, Automated Leak Detection, Identity Protection.

Abstract: Leakage of identity data is a precursor of identity theft in the Internet. Prevention measures are neither established to counteract identity theft nor is there any effective way to inform affected subjects after identity leakage has been discovered. To build an identity theft early warning system, it is crucial to find evidence of identity leakage that happened in the past. News sites in the Internet regularly report about organizations suffering from data leakage. Those leaked data mostly contains member, customer or employee databases including private information. This paper presents a framework that automatically crawls and classifies news articles with respect to identity data leakage. The framework is designed to monitor an arbitrary set of websites and to extract corresponding articles. The articles found are provided to analysts and security researchers with extracted information about the covered leaks. This lowers the amount of work that is necessary to stay up to date regarding leaks of identity data. The developed framework is a proof of concept and a foundation for further projects aiming to proactively warn affected users.

1 INTRODUCTION

Millions of user credentials, such as passwords or credit card information are breached from Internet services. Regularly, the leaked sensitive information of individuals is abused by criminals for identity theft. After the criminal act, many credentials are still valid and leaks are sold to other criminals or published publicly accessible. Huge identity leaks result from attacks on services like social media platforms or online shops. Criminal hackers usually copy the user database partially or completely after they successfully broke into the target system. More and more web services implement secure techniques to store customer credentials. If implemented correctly, these techniques render resulting leaks more or less unusable. Unfortunately, many web services do not secure their customer information using state-of-the-art security means.

The criticality of each identity leak depends on multiple factors. One factor is the kind of information included in the leaked data. While some leaks only reveal the email address and a hashed password, others contain the full name, the postal address, the birth date and some payment information alongside the clear text password. Affected individuals can be harmed in multiple ways. The obvious harm is financial loss due to fraud based on identity theft. Additionally, there can be social or psychological harm. At least, if a leak solely contains the email address and a hashed password, this implicitly discloses the usage of the certain web service by the affected individual.

Leaks are traded, distributed or sold on different kind of underground platforms. Many leaks circulate through these platforms over years before an affected individual recognizes the own involvement. To protect individuals against possible consequences resulting from identity leakage, there are several web services aiming to inform users about their identity leak status. Internationally, *Have i been pwned* is the most common service in this area (Hunt, 2017). However, there are some other services sharing the same intent: *HPI-Identity-Leak-Checker* (Hasso-Plattner-Institut für Digital Engineering gGmbH, 2017), *Uni-Bonn Leak-Checker* (University of Bonn, 2019). One major problem in operating such a service is the extensive necessity of computational and storage resources. To get access to relevant information, analysts manually search underground platforms in the Internet to identify newly available identity leaks. This is a very time consuming task, that needs to be done steadily to ensure the recency of the own leak

Malderle, T., Knauer, S., Lang, M., Wbbeling, M. and Meier, M.
Track Down Identity Leaks using Threat Intelligence.
DOI: 10.5220/0008963000960106
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 96-106
ISBN: 978-989-758-399-5; ISSN: 2184-4356

database. It seems to be impossible to fully automate this task, as most underground platforms protect their infrastructure against automated processing. Common hurdles are security mechanism like CAPTCHAS or different kind of social interaction.

Besides underground platforms, new leaks are reported by different sources, whereby the article contents differ significantly. Analyst need to subscribe to virtually every possible news service or blog to stay up-to-date. Nevertheless, relevant information regularly gets lost in the huge amount of available data. To decrease the time for scanning the platforms, it would be helpful for the analyst to have more specific information. Knowledge about a recently breached web service or company and a possible name of a leak might help to find the leak in the Internet. This allows to search systematically for this leak instead of crawling the whole Internet without a specific target. Therefore, an automatic filter is needed to filter out the relevant information only.

The work presented in this paper tries to close this gap. The main contributions are the steady monitoring of suitable sources and the rapid extraction of relevant information from the articles crawled from these sources to support security analysts. The required filter is realized with machine learning techniques. This filter is integrated in a system that assists security analysts monitoring relevant news services and being notified when a new leak occur. The developed framework aims to permanently monitor relevant sources, identifying news dealing with identity leaks by leveraging natural language processing and a trained support vector machine for the selection process. To further decrease the amount of data served to the security researcher a novel mechanism for determining the criticality of a leak is introduced, structuring the selected data in a usable web interface.

This paper is structured as following: Section 2 presents related work out of the categories *identity leakage* and *information retrieval*. Section 3 describes how the news articles are gathered. Their classification with a SVM into the groups *leak related article* and *unrelated article* is topic of section 4. Subsequently a comprehensive analyses will be performed on the leak related articles to extract needed features (section 5). Section 6 shows the entire concept of the evolved system and in section 7 all results will be summarized.

2 RELATED WORK

In computer science identity leakage, information retrieval and document classification are well known research fields. For a better understanding, the different research areas and their interaction for the designed system are presented in this section. The following section will focus on identity leakage and the section afterwards will outline the research field of information retrieval and document classification.

2.1 Identity Leakage

Identity leakage is a research field which investigates issues around users and their digital personal data. There are few subareas exploring identity leakage from different perspectives. Mainly, it can be divided into the technical and the human part. The technical part evaluates the leakage itself, detection of identity leakage and the protection against it (Thomas et al., 2017; Thomas et al., 2019; Gruss et al., 2018). The human part is about the humanly factors that can also be a reason for identity leakage like using weak or reused passwords (Mayer and Volkamer, 2018; Wash et al., 2016; Pearman et al., 2017; Stobert and Biddle, 2014; Stobert and Biddle, 2018). From the viewpoint of security researchers, the preventive and reactive countermeasures are examined which are supposed to secure the affected user and the compromised infrastructure.

A reactive countermeasure is described by DeBlasio et alii. They modelled a detection service for external web services (DeBlasio et al., 2017). This service registers honey pot accounts at the observing web services by using fake email accounts of a self-operating email server. The email mailbox has the same login credentials as used for the web service. As soon as someone authenticates himself with the correct password at such a mailbox the service knows that the login credentials are stolen from the observed web service (DeBlasio et al., 2017). The second viewpoint is the perspective from the human computer interaction. There are multiple studies on how users choose their passwords. Around 33 % of users reuse their passwords (Han et al., 2016; Subrayan et al., 2017) and if the password is not reused it is common to add a prefix or suffix, which leads to multiple similar passwords for one user (Han et al., 2016). Additionally, it is examined how to warn affected users about their involvement in an identity leak (Malderle et al., 2019).

2.2 Information Retrieval & Document Classification

While document classification is the umbrella term for operations classifying documents into different classes algorithmically, information retrieval de-

scribes the extraction of specific information from a big and or unordered source. Document classification is a very broad term, summing up whole research topics like image recognition or natural language analysis, including the classification of textual data, which is relevant to our research goal. When categorizing documents there are different ways to assign a category. It could be based on the documents' subject, metadata or other attributes. Hereby, the selection of suitable attributes is crucial for a sophisticated categorization as well as for further tasks like grouping. Automated information retrieval systems are used to reduce what has been called information overload. The most visible information retrieval systems are web search engines like Google and projects like Thesaurus (Dictionary.com, LLC, 2019; Aitchison and Dextre Clarke, 2004).

However, there are also more focused fields of information retrieval. Even in the scope of threat intelligence systems there are research works. Nunes et al. developed a threat intelligence system which gathers information about new malware and cyber-attacks by analyzing hacker forums and special marketplaces (Nunes et al., 2016). With a machine learning model their system gathers around 300 cyber threat warnings per week. Benjamin et al. additionally track and analyse the conversations in hacker IRC channels to detect potential future threats (Benjamin et al., 2015). Very similar to this research, Grisham et al. are using neural networks to identify information about new mobile malware in hacker forums (Grisham et al., 2017). All these works download content from different hacker forums. The problem is that the hacker forums are secured against automatic crawling by using different techniques like CAPTCHAS, invitational logins and so on. Wiliams et al. design a new approach to bypass these anti-crawling countermeasures by developing a specialized forum crawler (Williams et al., 2018). It is also examined how textual information from common threat intelligence sources can be analyzed for further automatic processing. Husari et al. evolved an ontology which can describe the context and the actions of cyber attacks (Husari et al., 2017). In addition they developed a new approach for text mining by using natural language processing and techniques from information retrieval to generate information which can be represented in their ontology. Furthermore, their tool allows to export this information to standard data formats for threat intelligence like *STIX 2.1* (Husari et al., 2017).

3 ARTICLE GATHERING

For gathering, suitable sources need to be identified. While there is a great amount of news producing sources like analog and digital media, the scope needed to be narrowed for effective sourcing. To limit the amount of possible sources to those being easily accessible by our framework, the following constraints were made:

Media. To enable automated collection and evaluation of the articles, the sources were limited to digital media.

Language. Regarding the current preference of English in the field of natural language processing as well as security research and computer science in general we focused on this language, based on the assumption that a relevant article about a new leak is going to be be published in English, at least as translation.

Accessibility. The source must be freely accessible and not hindered by limitations like pay walls, captcha-queries or censorship. While this form of technical hurdles could be overcome, it bears additional cost, not benefiting this proof of concept.

Significance. This quality can be acquired by having a significant scope or having specialized in IT security, ensuring a good coverage or high specificity.

Following these constraints, the chosen list contains sector specific websites like *security week*, *the hacker news* or *threat post* as well as popular international sites like *the guardian*. The complete list consists of: *Comodo* [1], *GBHackers* [2], *HackRead* [3], *Help Net Security* [4], *Infosecurity-Magazine* [5], *Security Gladiators* [6], *Security Week* [7], *Techworm* [8], *The Hacker News* [9], *Threat Post* [10], *The Guardian* [11], *Information Week* [12], *Naked Security* [13], *Trendmicro* [14],

[1] https://blog.comodo.com
[2] https://gbhackers.com/category/data-breach/
[3] https://www.hackread.com/hacking-news
[4] https://www.helpnetsecurity.com/view/news/
[5] https://www.infosecurity-magazine.com/news/
[6] https://securitygladiators.com/internet-security-news/
[7] https://www.securityweek.com
[8] https://www.techworm.net
[9] https://thehackernews.com
[10] https://threatpost.com/blog/
[11] https://www.theguardian.com/international/
[12] https://www.informationweek.com/
[13] https://nakedsecurity.sophos.com/
[14] https://www.trendmicro.com/vinfo/us/security/news/

Cyberdefense Magazine [15].

For each of these sources a crawler was designed. The crawlers are adjusted for each source, based on the framework *scrapy* (Scrapinghub, 2019). A manager module is managing the spider swarm and crawling every source at a configurable frequency, favorably multiple times a day, downloading the new published articles. After downloading, the articles get tokenized, tagged with source and timestamp and are stored in a database. The biggest challenge besides selecting useful sources was to write crawlers that would populate the database with homogeneous and sanitized data, following hyperlinks where it is helpful and skipping them everywhere else. The rules modelling the crawlers behavior were crafted for every source after a manual analysis of the sources' structure.

The crawling of these news outlets via the aforementioned spiders yielded a total of 52382 articles, which were saved in a mongoDB. The amount of articles from different sources varied greatly, ranging from 1646 articles from cyberdefense.com to 15163 articles from securityweek.com. The oldest crawled articles dated back to May 2007, resulting in an average of 12 articles per day for a time span of 12 years.

4 ARTICLE CLASSIFICATION

A linear support vector machine (SVM) is a linear machine learning model, which is widely used in text categorization. It has several advantages, such as a good generalization in data with wide feature sets (Joachims, 1998). This characteristic pays off in the analysis of textual data composed of thousands of different words. Therefore a linear SVM is well-suited for the classification of news articles and leak news articles.

The classification process had two main challenges: On one side finding a way to distinguish articles that deal with *leaks* semantically from those who don't. On the other side decide, whether an article about *leaks* is about the form of digital *identity leakage* or about unrelated forms of leakage like oil leakage, or broken water pipes, to just name a few.

The articles in question obviously consist of a title and the actual content. A skimming through different news sources lead to the impression that, for a human reader the classification as leak/ non leak is possible for most articles by only looking at the title. Since the length of the typical article title is only a fraction of the length of the typical article content, the title-only analysis consumes fewer computational resources than the analysis of the entire article. Because of that it seemed reasonable to test if a classification of articles based only on the title can lead to good results. In addition, the classification of the articles based on the whole article is tested.

[15]http://www.cyberdefensemagazine.com/category/news/

4.1 Building a Training Set

In order to build a machine learning classification algorithm, it is unavoidable to manually classify a sufficient number of articles in order to form a training and a test set. To facilitate this task, a GUI was built to easily flag an article as leak or non-leak by a single click. A combined filtering of the articles regarding date, source and keywords allows to group similar articles and manually classify them reasonably fast. After initially classifying a few hundred articles in the described way, a first version of the optimizer, which is described later, was tested on non-classified articles. The articles which were predicted as leak related were then again manually checked to broaden the training set. Since the overall share of leak related articles is relatively small, this was a feasible task. The repeated application of both approaches lead to a total of 15217 classified articles forming the test set of which 1996 were categorized as leak related. Due to the described approach, the percentage of leak articles in the training set is considerably higher than in the complete data set. Overall, the leak related articles made up 3.8 % of all articles collected.

4.2 Text Pre-processing

Before being able to be classified by the SVM, the titles, and in the second stage the text bodies, of the manually classified articles need to be pre-processed. This process consists of a tokenization and pos-tagging of the text with the help of the Natural Language Toolkit (NLTK). After this step, the tokenized and tagged representation of the articles were used as feature sets for the consecutive steps. This representation is characterized by the reduction of raw data by deleting stop-words, conjunctions and alike and attaching categories like "nouns", "numeric data", "verbs", etc. to the features.

4.3 Learning

Afterwards, the data is vectorized by SKLearns CountVectorizer, split in a training and a test set and finally processed by the SVM via the Python library *scikit-learn* (scikit-learn, 2019). To find the best parameters for the optimizer a grid search was per-

formed. The best found parameters lead to the following results, looking at the title only approach: Classification of the articles as leak related was done with a precision of 0.80 and a recall of 0.77, classification as non-leak related with a precision of 0.96 and a recall of 0.97. These numbers, although not completely terrible, do warrant a further investment of time and resources into the analysis of the complete content of the articles to improve the recognition of articles dealing with leaks.

Concerning the full text body approach the lexical analysis and classification pipeline is improved, consisting of a tf-idf step and a classification step with a trained SVM model. To improve the performance of the already existing pipeline, it gets retrained with more data points. For further information on the technical details consult the previous work on automated leak classification. This larger feature set showed significant improvements over the approach using the titles only. While consuming more computational power in the training and the service stage, the precision of retrieving leak related articles came in at 0.876 and the recall showed rates above 0.95. The final recall value depends on how sharp the term ”leak related” is defined and how borderline cases are tagged.

5 FEATURE EXTRACTION OF LEAK NEWS ARTICLES

When dealing with articles about leaks it is necessary to extract vital information about the data breach: How many users were affected? Which kind of identity data was stolen? When was the breach?

5.1 Extract Breached Service Detection

For extracting the breached web service from a leak news article it is necessary to specify how that information can be presented in an article. The obvious part of this specification is the position in the article. Consisting of the three aspects title, text and tags, the collected articles referenced the desired information partly in their title, partly in their text, sometimes both, and sometimes as addition in the tags section. Regarding that there are no cases which had the information in the tags but in neither of the other fields, the location for searching the breached service was narrowed to the text and title fields. On these fields, which were tokenized and saved by the crawler, a part of speech tagging is performed. This process includes the selection of all tokens representing nouns and reducing the selection by comparing the tokens to the Alexa 1 Million (Amazon, 2019) lists, containing the million biggest web services. This results in an entity recognition process which produces a list of possible breached sources mentioned by the article. To extract the desired service a bunch of filtering and weighting steps are taken. At first repeatedly named entities are weighted higher, representing their frequency in the article. Second, known correlating names, for example typical email providers like *yahoo*, *gmail*, *gmx*, or often mentioned security researchers like *Troy Hunt* (and his site *have i been pwned* (Hunt, 2017)) are weighted lower. These entities represent a major amount of the false positives returned by the entity recognition process. It was still a design goal not to filter them out strictly, because even if they regularly aren't the source of an identity leak, they could be. In the end the breached service recognition yields a (sorted) list representing the most probable sources for the leak in question.

5.2 Extract Affected Identities

Another critical information for evaluating the potential harm of an identity leak as well as estimating the best way for a victim warning, is to know how many identities are affected. This information is - if present at all - included in the text (possibly also the title) of the article in numeric or written-out/spelled-out form. Leveraging the advantages of the previously done speech tagging, the tokens, representing these number associated features, are extracted. Appended to a list the numbers extracted from the title are weighted three times as heavy as numbers originating from the text. In the idea that another number, maybe referencing an event the article author wanted to show similarities to, would need to be repeated more than the usual reference does (1 or 2 times) to overwrite the impression from the title. At the end the most frequent (and therefore most weighing) numbers are shown to the analyst.

5.3 Extract Leaked Data Type

Furthermore it is important to know what type of identity data has been leaked. While a leakage of any data type is undesirable, especially when taking into consideration recent incidents like the 'politician doxing' in Germany in early 2019 (Dobuch, 2019), critical information like financial data, clear text passwords or breaches at sensitive services like adult sites can cause and have caused more harm than others (Thomson, 2015). To filter for these information the natural language processing extracts all words known to be affiliated with these groups and stores them as metadata in the database. These extracted lists are

used in the threat assessment step to determine the potential harm.

5.4 Threat Assessment

Mandatory tasks for providing meaningful threat assessment and warning services are to determine the actuality and relevance of the leak. Therefore it is desirable to qualify an article as either referring to a current, yet unknown breach or an old, already known one. Known in this context can mean known by this frameworks database - *called internal matching*.

5.4.1 Internal Matching - Clustering

Internal matching describes the comparison of new leak articles with ones stored in the database. Because of the in-homogeneous nature of journalistic work this comparison can not take place at document level by for example comparing their hash sums. In general, two articles vastly differing in style, written by different people from various news sites can very well be about the same leak, whereas two very similar articles could deal with different ones. To overcome this challenge the internal matching is realized by creating a network of leak clusters. They are developed by grouping up the articles with similar breached services and filtered by time. For getting useful clusters the process has been redone with several, shrinking time windows, observing the resulting clustering of known leak articles. This showed a typical time span of less than a week after the disclosure of a leak, during which the reporting takes place. An additional effect of using time windows as a feature is the separation of survey articles which happen to deal with the same services and affected user counts as the actual leak articles, but are regularly published with a temporal distance and often contain more detected services than matching to a single leak.

5.4.2 Topicality

Another important information about an article handling a leak is how old the leak is. This can reference the time when the source service was breached, the point when the leaked data was published, or the point when the media began reporting. These points can fall close together but do not have to. Typically data gets stolen and is sold indefinitely later, a timespan which can be as long as several years. Concerning media coverage, it regularly starts hours to a few days after the leak was published on sites like *pastebin.com*.

Because of the uncertainty associated with information about the breach taking place and the first occurrence on paste sites, we are focusing on the timestamps related to the articles dealing with the respective leaks.

Dealing with this process, it was evident that the report duration is significantly shorter than the span between breach and becoming common currency. While within the first 48 hours about 80 % of the articles concerning the same leak are published, it was observed that 99.4 % of all articles about the same leak are published within 6 days, staying just under a week of reporting time. This observation influenced the design of the clustering process.

5.4.3 Threat Level Rating

To filter the amount of data, the analyst has to recognize regularly, it is important to assess the relevance of the leak news. In this regard relevance means the level of potential harm that could be associated with this new leak.
Because yet no perfect model for this application exists, a constructive approach was met. The model was designed with a security researcher as operator in mind and based on the collectible information, accessible without privilege or financial expense. Figure 1 shows the process for evaluating the threat level of a detected leak. Ensuring a reliable metric without the need for future expansion in scale we decided to favor a usable approach, scaling between 1 *smallest potential threat*, and 9 *biggest potential threat*.

When an article was classified as a leak and the extraction of the supposed service and affected user count has been done, the leak article is integrated in the cluster model to decide whether the reported incident is already known or not. At this point additional external matching services can be applied. As figure 1 shows, the Threat Level is set to one if the article is known and set to five otherwise. This step was designed with a frequent usage of this tool in mind. This ensures that an incident already recognized by the admin will not be seen as an imminent threat thus not reported by the notification system. Additionally, this ensures that an already known leak, irrespective of the following evaluation steps like affected user count or stolen data-type, is always valued below 5, keeping the web-interface uncluttered. This allows a selective view on more relevant data for the operator in the web interface. To get to a more meaningful metric the threat model weighs in several aspects, represented by the questions in 1. If answered positively the threat level is incremented.

Service Reputation. Is the breached Service in the *Alexa Top 1k* list, representing a big, possibly world wide acting business?

Outstanding Affected User Count. Are more users

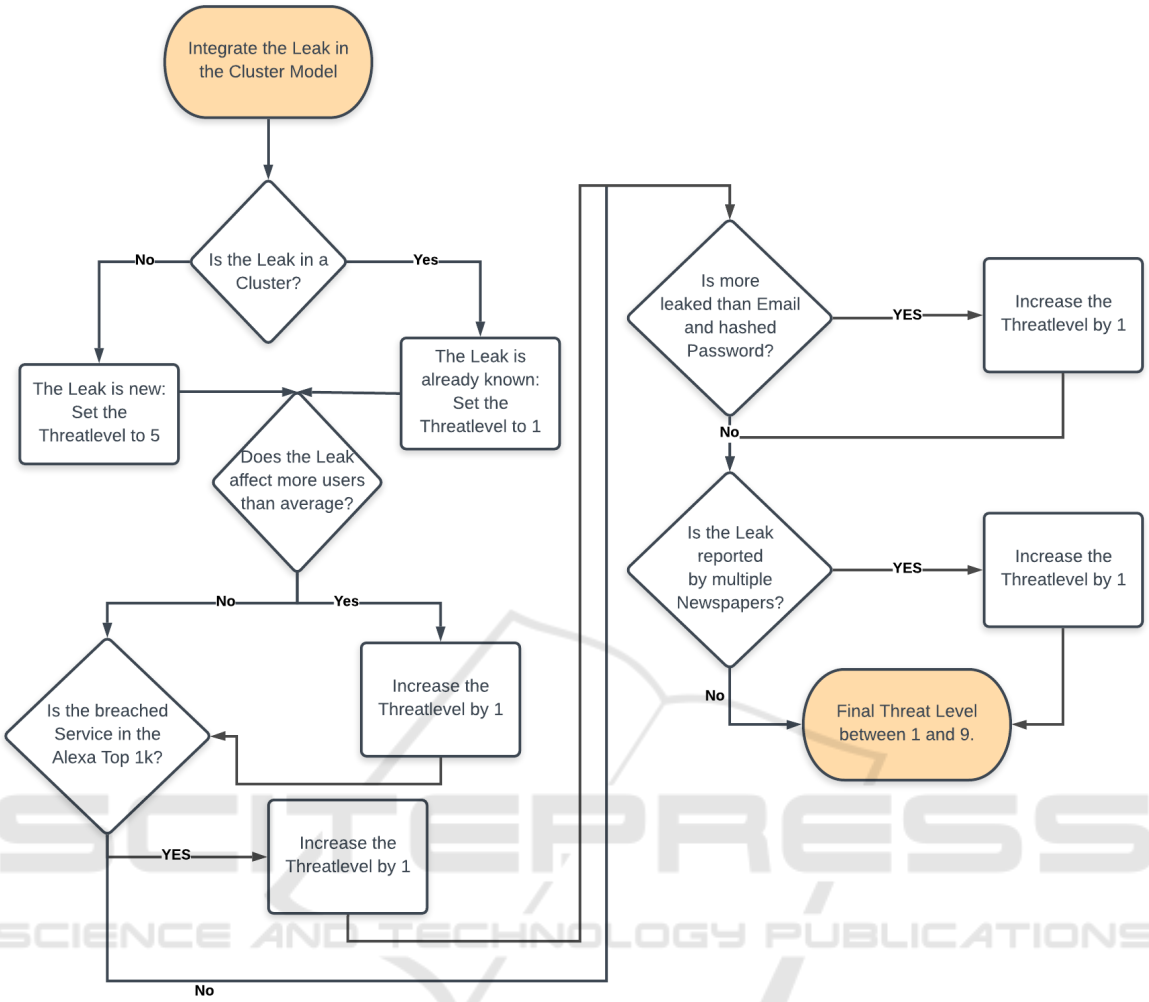

Figure 1: Threat Level Rating Process.

affected than in the average leak?

Valuable Information Leaked. Is more data leaked than email addresses and hashed passwords? For example clear text passwords, credit card data, or vast amounts of private data like addresses, telephone numbers.

Popular by Newspapers. If the article is handling a new, jet unknown leak: Did more than three newspaper articles came up within 48 hours? Then there seems to be a special interest in this particular leak, so the analyst should be informed.

The model waives to differentiate between differently answered threat level questions resulting in the same score, providing an easy to understand metric for the operator, keeping all additional information in its linked data.

6 CONCEPTUALIZATION OF THE ENTIRE SYSTEM

This section sketches the concept of the leak-news detection framework. Concentrating on the implementation, the framework consists of three stages forming a processing pipeline. Stage one handles the download and categorizing of articles, using *scrapy* crawlers (Scrapinghub, 2019) and a trained linear SVM. If stage one categorizes an article as dealing with identity leaks, it starts the parsing process. The Parser, divided in several modules, composes stage two. Once stage two gets used the program flow always triggers stage three, consisting of the threat level assessment and according reactions from displaying the information in the web-interface to immediately write an email to its supervisor. This process, with additional steps like the evaluation of potential articles sources

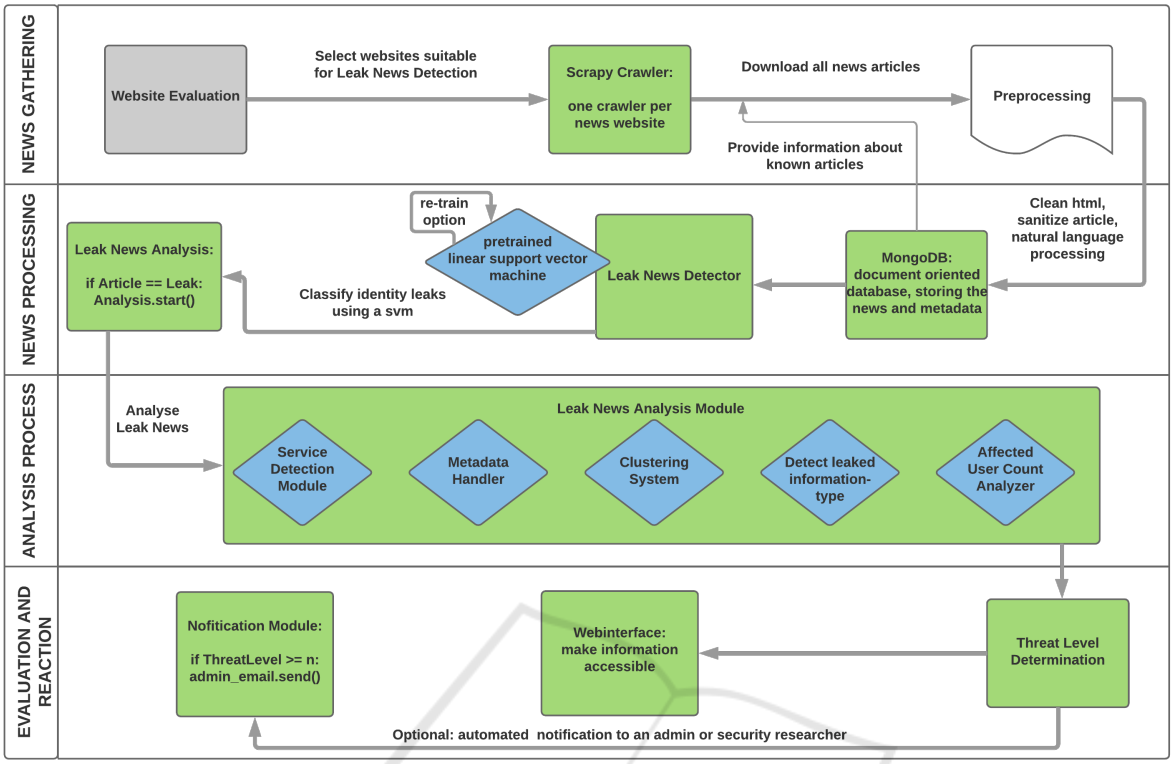

Figure 2: Framework Pipeline.

or the retrain option for the SVM is shown in figure 2. Figure 3 shows the GUI which is used to facilitate the manual classification of articles. Articles can be filtered by date, source or keywords. The background color indicates whether an article was manually classified either as leak-related (green), or non-leak related (red). A true or false flag in the 'Pred' column signifies that an article was part of the test set and predicted to be leak related or not.

7 EVALUATION

To evaluate the usefulness of the framework, the classification process, the extraction of leak information and the threat assessment are looked at separately.

7.1 Information Extraction

Because of the information retrieving nature of this process the metrics precision and recall are also suitable. For the extraction process recall describes the amount of cases in which the correct affected user count and breached service could be detected compared to all cases. Precision describes how many false/unnecessary/unwanted information was retrieved as well. The return of the extraction process yields in a much lower volume than a complete leak article offers. This results in a lower weighting of the precision metric. Supposedly a security researcher would be able and interested to get a list with three to four services, from which one is the service which got breached, linking to a single article explaining the case. In contrast he would probably unsubscribe a *leak news detection service* which would send four articles, with only one handling news. Trusting on the filtering skills of the security researcher we regard recall the main Q-degree for this module.

7.1.1 Breached Service Extraction

Looking into the extraction of the service or company that had suffered a data breach, there are two recall metrics we can look into. On one hand the articles containing the name of the service in their text could be successfully parsed with a recall of 0.95. The resulting margin of 0.05 are companies with names not recognized by the pipeline. The main reasons for this are a too unusual name, not identified by the NLP step in the first place, or company names that are too close to normal English words, also skipping the NLP process. Additionally, services to small to appear on the Alexa 1 million (Amazon, 2019) list are excluded as

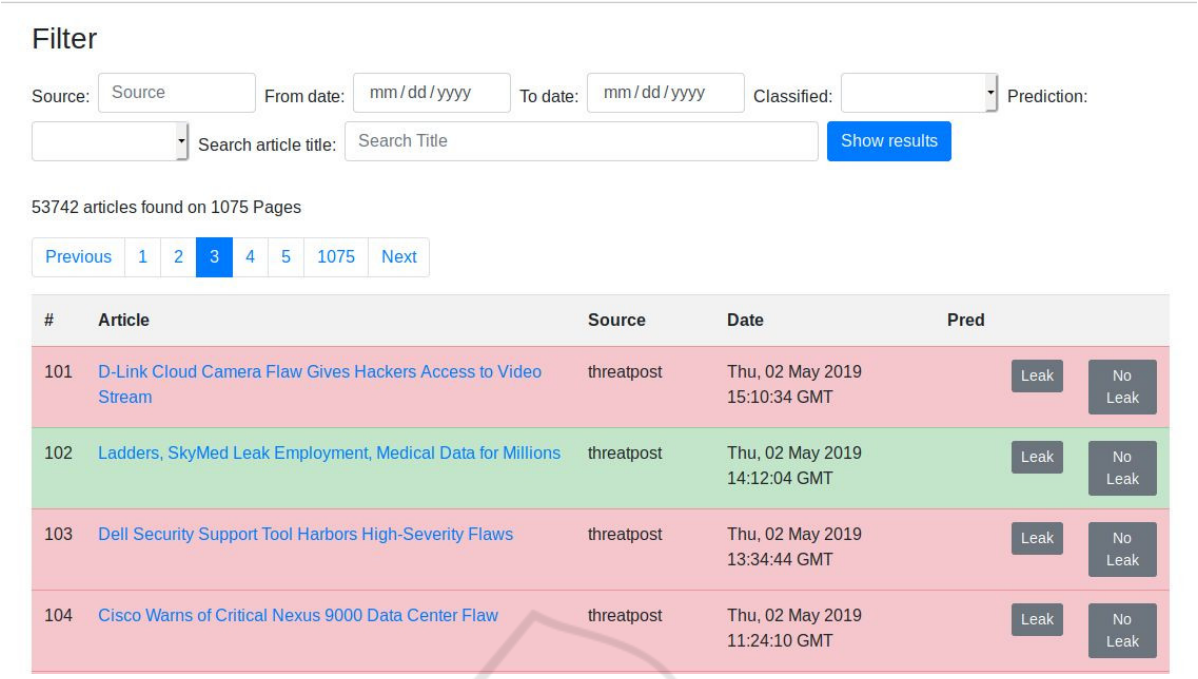

Figure 3: GUI.

well.

On the other hand when just looking on the extracted services for all classified articles we find a recall of 0.89 because not all articles reporting about a hack or credentials appearing on marketplaces name any affected company, resulting in a lowering of the overall recall. In regard of the fact that this decrease is not rooted in the framework itself but in the underlying articles, the recall of 0.95 should be looked at to assess the performance of the extraction feature. It should be noted that the process to determine this metric did not make a difference whether the extracted entity was a service or a company as long as a security analyst could use the information as well. As example, many news articles reported about hacks of *Steam* and the *PlayStation Network*, whilst some others reported about leaks at *Valve* and *Sony*, both information is correct, and viewed equal.

Looking at the accuracy of the weighing process, it is the best metric for this case. The likelihood of having the breached service in one of the first three position of the potential service array is about 70 %.

7.1.2 Affected User Count

The affected user count can be parsed in 95 % of the cases, resulting in a high recall of 0.95. The remaining cases do not give affected user counts at all, or are survey articles resuming about too many high user counts for the framework to keep track off. The accuracy is determined as for the breached service detection above, closely bound to the position in the returned feature list. Having the correct affected user count in one of the first three positions of the potential user count array, the likelihood is about 85 %. Because there is no fixed position in the text stating the correct count, vs. additional affected user counts, referring to different leaks, information like these could not be ignored easily in the natural language processing, but is considered future work, leaving room for further improvement.

7.1.3 Information Type Detection

As we are interested in the possible impact factor of the leaked data we cared about whether the data type is member of a group of especially dangerous material, containing financial data like credit card numbers, clear text passwords, or socially sensible information like breaches at adult websites for example. This process is implemented as a list matching and retrieves the pre-selected terms reliably. The information about the leaked data, disregarding emails as a type, was given in less than 20 % of the cases. In the scope of this project it was not possible to assess whether this was due to less sensible data being compromised or to just a few newspapers reporting.

7.2 Threat Assessment

Assessing the performance of the computed threat level is quite difficult as it contains a subjective com-

ponent, as its performance would be measured in the *usability* for the security researcher using our framework. The best metric would be a usability study with for example a derived SUS questionnaires and a group of security researchers as participants. Sadly this was not feasible in the scope of this project, so we concentrated on the constructive approach met in designing the assessment module, regarding a final evaluation also as future work. By design the framework concentrates on a few crucial features to determine the harmfulness of a leak. Most users should agree that a leak is more dangerous if more people than average are affected. Also it is common sense that a leakage of clear text passwords or credit card information is more critical than leaked password hashes. The consideration whether some data types in our *critical* category are more harmful than the other ones is not part of this project, but could be researched in future projects. It was the priority of the found framework to clearly identify a leak that is definitely harmful and give a warning via the web-interface and optionally warn or inform its operator by email. Being assembled in a pipeline, each stage relies on the work of its predecessors. If no affected user count could be detected, it could not be compared to the average affected user count. Regarding that we implemented a medium threat level area in which each user has to decide on its own whether it is worth to look further into it or not, configurable by an email at threat-level threshold variable. This short coming is not rooted in the process itself, but determined by the limited information provided by the news articles in the first place.

8 CONCLUSION

Login credentials are crucial information about digital identities of individuals. Attacks on service providers regularly bring these identities into the unauthorized hands of criminals. The information is then used for identity theft or it is aggregated into larger collections and sold accordingly. This paper contributes with a new approach to identify digital identity leakage as foundation for a larger framework to proactively inform affected individuals. Therefore, a wide variety of newspapers and blogs are crawled on a regular basis. After the identification of relevant articles, the substantial information is extracted and forwarded to a security analyst. It is shown that relevant articles can be reliably classified by in depth text analysis. Furthermore, it is possible to extract required information to identify the source of the leak and the amount of identities affected. Running as a service, it steadily monitors the Internet for related information with low operational cost. Even though the described software cannot substitute security analysts, it decisively supports them in tracking down identity leaks, be they small or huge collections of stolen identities.

REFERENCES

Aitchison, J. and Dextre Clarke, S. (2004). The thesaurus: a historical viewpoint, with a look to the future. In *Cataloging & Classification Quarterly*, volume 37, pages 5–21.

Amazon (2019). Alexa top million. https://aws.amazon.com/de/alexa-top-sites/. Last accessed 2019/12/17.

Benjamin, V., Li, W., Holt, T., and Chen, H. (2015). Exploring threats and vulnerabilities in hacker web: Forums, irc and carding shops. In *2015 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 85–90.

DeBlasio, J., Savage, S., Voelker, G. M., and Snoeren, A. C. (2017). Tripwire: Inferring internet site compromise. In *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, pages 341–354, New York, NY, USA. ACM.

Dictionary.com, LLC (2019). Thesaurus.com - synonyms and antonyms of words. https://www.thesaurus.com. Last accessed 2019/12/17.

Dobuch, G. (2019). 20-year-old german hacker confesses in doxxing case. https://www.handelsblatt.com/23841212.html. Last accessed 2019/12/17.

Grisham, J., Samtani, S., Patton, M., and Chen, H. (2017). Identifying mobile malware and key threat actors in online hacker forums for proactive cyber threat intelligence. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 13–18.

Gruss, D., Schwarz, M., Wübbeling, M., Guggi, S., Malderle, T., More, S., and Lipp, M. (2018). Use-after-FreeMail: Generalizing the use-after-free problem and applying it to email services. *ASIACCS 2018 - Proceedings of the 2018 ACM Asia Conference on Computer and Communications Security*, pages 297–311.

Han, W., Li, Z., Ni, M., Gu, G., and Xu, W. (2016). Shadow Attacks based on Password Reuses: A Quantitative Empirical View. *IEEE Transactions on Dependable and Secure Computing*, X(X):1–1.

Hasso-Plattner-Institut für Digital Engineering gGmbH (2017). HPI Leak Checker. https://sec.hpi.de/leak-checker. Last accessed 2019/12/17.

Hunt, T. (2017). have i been pwned? https://haveibeenpwned.com. Last accessed 2019/12/17.

Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., and Niu, X. (2017). Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACSAC 2017, pages 103–115, New York, NY, USA. ACM.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *ECML*.

Malderle, T., Wübbeling, M., Knauer, S., and Meier, M. (2019). Warning of affected users about an identity leak. In Madureira, A. M., Abraham, A., Gandhi, N., Silva, C., and Antunes, M., editors, *Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018)*, pages 278–287. Springer International Publishing.

Mayer, P. and Volkamer, M. (2018). Addressing misconceptions about password security effectively. *ACM International Conference Proceeding Series*.

Nunes, E., Diab, A., Gunn, A., Marin, E., Mishra, V., Paliath, V., Robertson, J., Shakarian, J., Thart, A., and Shakarian, P. (2016). Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pages 7–12.

Pearman, S., Thomas, J., Naeini, P. E., Habib, H., Bauer, L., Christin, N., Cranor, L. F., Egelmany, S., and Forgetz, A. (2017). Let's Go in for a closer look: Observing passwords in their natural habitat. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 295–310.

scikit-learn (2019). scikit-learn - Machine Learning in Python. https://scikit-learn.org/stable/. Last accessed 2019/12/17.

Scrapinghub (2019). Scrapy. https://scrapy.org/. Last accessed 2019/12/17.

Stobert, E. and Biddle, R. (2014). The Password Life Cycle: User Behaviour in Managing Passwords. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, volume 10, Menlo Park, CA. USENIX Association.

Stobert, E. and Biddle, R. (2018). The password life cycle. *ACM Transactions on Privacy and Security*, 21(3).

Subrayan, S., Mugilan, S., Sivanesan, B., and Kalaivani, S. (2017). Multi-factor Authentication Scheme for Shadow Attacks in Social Network. *2017 International Conference on Technical Advancements in Computers and Communications (ICTACC)*, pages 36–40.

Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A., Margolis, D., Paxson, V., and Bursztein, E. (2017). Data Breaches, phishing, or malware? understanding the risks of stolen credentials. *Proceedings of the ACM Conference on Computer and Communications Security*, pages 1421–1434.

Thomas, K., Pullman, J., Yeo, K., Raghunathan, A., Kelley, P. G., Invernizzi, L., Benko, B., Pietraszek, T., Patel, S., Boneh, D., and Bursztein, E. (2019). Protecting accounts from credential stuffing with password breach alerting. *28th {USENIX} Security Symposium, {USENIX} Security 2019, Santa Clara, CA, USA, August 14-16, 2019.*, pages 1556–1571.

Thomson, I. (2015). More deaths linked to ashley madison hack as scammers move in. Last accessed 2019/12/17.

University of Bonn (2019). Uni bonn identity leak checker. https://leakchecker.uni-bonn.de. Last accessed 2019/12/17.

Wash, R., Rader, E., Berman, R., and Wellmer, Z. (2016). Understanding Password Choices: How Frequently Entered Passwords Are Re-used across Websites. *Twelfth Symposium on Usable Privacy and Security*, (Soups):175.

Williams, R., Samtani, S., Patton, M., and Chen, H. (2018). Incremental hacker forum exploit collection and classification for proactive cyber threat intelligence: An exploratory study. In *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 94–99.