Title	Towards an Efficient Log Data Protection in Software Systems through Data Minimization and Anonymization
Authors(s)	Portillo Dominguez, Andres Omar; Ayala-Rivera, Vanessa
Publication date	2019-10-25
Publication information	Juárez-Ramírez, R., Fernández y Fernández, C., Jiménez, S., Ramírez-Noriega, A., Pérez González, H., Licea Sandoval, G., Guerra-García, C. (eds.). 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT), Mexico City, Mexico, 23-25 October 2019: Proceedings
Conference details	The 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT), Mexico City, Mexico, 23 -25 October 2019
Publisher	IEEE
Link to online version	http://conisoft.org/2019/
Item record/more information	http://hdl.handle.net/10197/12346
Publisher's statement	© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/conisoft.2019.00024

Towards an Efficient Log Data Protection in Software Systems through Data Minimization and Anonymization

A. Omar Portillo-Dominguez, Vanessa Ayala-Rivera
Lero@UCD, School of Computer Science,
University College Dublin, Belfield, Dublin 4, Ireland.
Email: {andres.portillodominguez,vanessa.ayalarivera}@ucd.ie

***Abstract*—IT infrastructures of companies generate large amounts of log data every day. These logs are typically analyzed by software engineers to gain insights about activities occurring within a company (e.g., to debug issues exhibited by the production systems). To facilitate this process, log data management is often outsourced to cloud providers. However, logs may contain information that is sensitive by nature and considered personal identifiable under most of the new privacy protection laws, such as the European General Data Protection Regulation (GDPR). To ensure that companies do not violate regulatory compliance, they must adopt, in their software systems, appropriate data protection measures. Such privacy protection laws also promote the use of anonymization techniques as possible mechanisms to operationalize data protection. However, companies struggle to put anonymization in practice due to the lack of integrated, intuitive, and easy-to-use tools that accommodate effectively with their log management systems. In this paper, we propose an automatic approach (SafeLog) to filter out information and anonymize log streams to safeguard the confidentiality of sensitive data and prevent its exposure and misuse from third parties. Our results show that atomic anonymization operations can be effectively applied to log streams to preserve the confidentiality of information, while still allowing to conduct different types of analysis tasks such as users behavior, and anomaly detection. Our approach also reduces the amount of data sent to cloud vendors, hence decreasing the financial costs and the risk of overexposing information.**

***Keywords*-Software Engineering, Privacy, Security, Anonymization**

I. Introduction

The introduction of new and more strict privacy protection laws, such as the California Consumer Privacy Act 2018 [1] and the General Data Protection Regulation 2016/679 [2] (hereinafter GDPR) have brought multiple obligations to companies [3]. One of the most important ones is accountability, which involves taking responsibility for complying with the GDPR and being able to demonstrate it. The collection of log and event data as a means to demonstrate compliance has become a standard. This is because it provides a trail of the different activities conducted within a company and maintains records of them. Additionally, log data is a valuable source of information to gain insights into the operational environment of a company such as monitoring the IT infrastructure and software systems' behavior [4].

However, log data needs to be handled properly as not only it can be exploited to breach into a system, but it may also contain information that is now classified as personal data under the GDPR. For example, usernames, passwords, IP addresses, cookies, protocols, email addresses, and other metadata which now fall within the scope of "online identifiers".

To prevent errors that may lead to privacy incidents, companies are required to put in place "appropriate safeguards" that protect all collected and/or generated data against unauthorized or unlawful processing throughout its lifecycle. For instance, Twitter and GitHub recently reported a bug in their internal logging systems where users' passwords were recorded in plaintext format [5], [6]. Although these events do not represent a formal data breach (as only a small number of employees had access to the exposed passwords in those logs), big security issues can occur if logs are not properly protected. Also, the liability of employees increases as they may qualify as "data processors" under the GDPR. Privacy risks are higher due to the fact that logs streams are often outsourced to cloud vendors to facilitate their management due to their volume or the lack of resources within a company [7].

GDPR promotes the use of technical measures such as pseudonymization and encryption as concrete and practical mechanisms to operationalize data protection (Art. 25 Data protection by design and by default). However, data anonymization (as we will refer in general to the set of de-identification techniques and their different variations) still brings many challenges for companies in order to effectively put it in practice [8]. Some of these challenges are the lack of easy-to-use and practical tools that implement concrete privacy-preserving techniques, the complex and expert knowledge involved in the anonymization mechanisms, the degradation of analytics utility after anonymizing data, and the worry of properly apply anonymization to prevent privacy leak incidents [9], [10].

In this paper, we examine the feasibility of using data minimization and confidentiality principles to protect log and event data that is outsourced to log management systems. We study the impact that these two techniques (i.e., data filtering and anonymization) have when conducting a set of

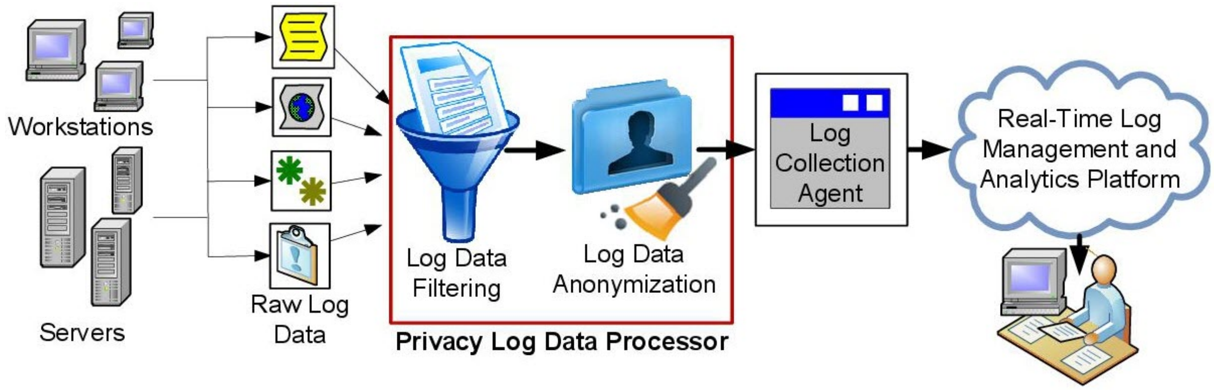

Figure 1. Proposed Solution in a Log Management System

analysis tasks typically performed on logs. More specifically, our contributions are the following:

1) We detail the use of automated approach (SafeLog) for filtering and anonymizing logs that contain different types of sensitive information and detail its interfacing with a cloud-based, real-time log-analysis platform.
2) We present a comprehensive practical evaluation of SafeLog, consisting of a prototype and a set of experiments to assess the tradeoff of privacy and utility in terms of the costs and benefits that the framework brings to different types of log analysis.

The rest of this paper is structured as follows: Section II details our solution; while Sections III and IV discuss the implemented prototype, the performed experiments and their results. Section V presents the related work. Section VI draws the conclusions and provides pointers to future work.

II. Proposed Approach: SafeLog

In this section, we describe our proposed approach: First, we provide the overview of the solution. Next, we describe in detail its main components. Finally, we conclude the section with a discussion of the supported anonymization/de-identification techniques.

A. Overview

The overview and scenario of our proposed approach are depicted in Fig. 1. Different types of log data are constantly generated within a company such as system logs, audit logs, application logs, configuration logs, and security logs. These logs are often analyzed by multiple teams within the company for distinct purposes such as evaluating if security controls are upheld by teams, verifying if applications are working properly, checking the events occurred during transactions, etc. To facilitate such analysis tasks, logs are gathered by a log collection agent and pushed periodically to a log management and analysis platform deployed in the cloud (e.g., DataDog [11], EventLog Analyzer [12], LogEntries [13]). Once in the cloud system, the analyst can conduct various tasks with log and event data. This process presents several issues for the new data protection by design and by default paradigm mandated by new privacy laws (e.g., GDPR) as confidential and sensitive information is overexposed to cloud providers personnel, IT support, or developers.

To address this problem, we propose to plug a new component, acting as a Privacy Log Data Processor (SafeLog), that serves as a proxy installed between the raw log data and the analyst in order to protect the data before sending it to the log collection agent. The aim of SafeLog is to make the log processing flow more efficient (by minimizing the amount of data that is sent to the cloud), as well as safer (by offering data de-identification solutions).

B. Architecture

The main components of SafeLog are shown in Fig. 2. The *filter* and the *anonymizer* act as data processors for the log streams. The *tools* are components that can be configured by the users (e.g., IT professionals) to customize the filtering and anonymization processes. These can be defined according to internal privacy policies of a company. The *instruments* influence the behavior of a tool component, they can be internal or external artifacts. In the following paragraphs, we describe in more detail the functionality of each component.

1) Filter: The filter component is motivated by how privacy laws have changed the way in which systems and applications were built in the past: from gathering as much data as possible, to minimize the processing of data to what it is required for a specific purpose. The filter is responsible for removing from logs the information that is considered redundant, trivial, obsolete, or not required for further processing. In this manner, the company preserves only "relevant" information such as "significant" events occurring within a data processing system. This filtering strategy brings various benefits to a company such as lowering cloud service expenses (in pay-per-use schemes by

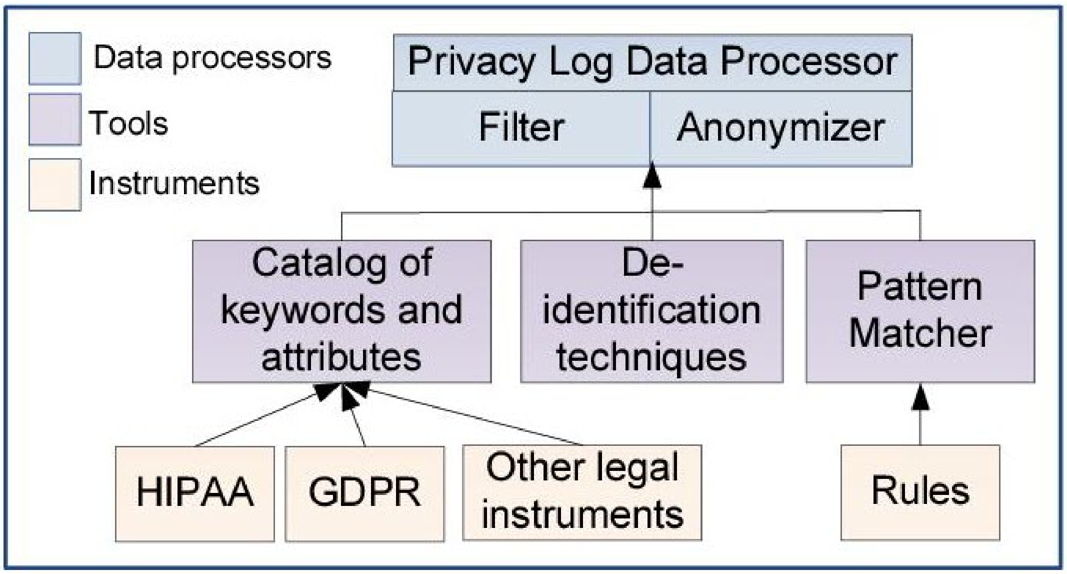

Figure 2. SafeLog's Core Components

reducing the amount of data sent to the cloud), decreasing the risk of data exposure (by only preserving data related to achieving certain goals), and improving the efficiency of data processing (by facilitating the correlation of data). It is worth highlighting that, in our scenario, the logs sent to the log management system mainly serve for business analysis purposes and are not related to digital forensics or audit procedures which are usually transferred to a centralized server used for cold storage. Such archived audit trails are tied to stricter conditions like completeness, integrity, and non-repudiation (i.e., demonstrate that data has not been tampered).

2) Anonymizer: The second main component of SafeLog is an anonymization engine which obfuscates the log data to protect their confidentiality when sending it to the log management system. This aligns with new privacy requirements stated in privacy laws which emphasize that only the personal data that is required for a specific purpose must be processed, this includes their accessibility (i.e., data must not be made accessible to an indefinite number of persons) and the extent of the processing. The application of anonymization techniques can help to fulfill this requirement by limiting the disclosure of the data and, depending on the degree of anonymization applied, can simplify or help to exempt responsibilities with the applicable privacy law. For example, when the anonymized data no longer allows identification of individuals to whom it relates, typically the privacy laws do not apply. More details about the anonymization techniques available in our solution are given in the following paragraphs.

3) Pattern Matcher: In order to apply any transformation to the log streams, these are parsed to find the messages that match a particular pattern (previously specified by users in the form of rules). The rules can be configured to match a specific string or indicated in the form of regular expressions, although the former would perform faster. Both, the Filter and the Anonymizer, utilize the Pattern Matcher to execute their specific functionality. In the case of the Filter, possible actions can be to drop messages meeting certain criteria or to ignore log events from a specific source. For instance, drop logs whose line starts with a specific date, a level of debugging (e.g., debug, info, error), or an application's name. In the case of the Anonymizer, the Pattern Matcher searches for sensitive data (currently manually indicated by the user) in order to transform it.

4) Catalog of Keywords and Attributes: The users can configure what is confidential to their company by setting up a set of keywords and attributes in a catalog for reuse. These sensitive terms can be aligned with the privacy policies and data classification schemes existing in the company. For example, there are different levels of confidentiality which can be situational specific to more generic ones. Depending on the business nature of the company, confidentiality levels can range from restricted, protected, and public, or pass through high-sensitive, low-sensitive, and nonsensitive. As long as there exists a scheme that is relevant, clear, and simple to the company. Moreover, depending on the type of data that a system processes, there are certain attributes that are sensitive by nature that must be protected against unwarranted disclosure. For instance, card holder data, protected health information, and student education records. This sensitive information is regulated by legal instruments such as the GDPR, HIPAA, FERPA, PCI, etc. which lists some attributes that need to be safeguarded, or

Table I
EXAMPLES OF PERSONAL IDENTIFIABLE INFORMATION IN REGULATIONS

HIPAA	
Name	Medical record number
Address	Health plan beneficiary number
IP Address	Account number
Fax numbers	Certificate or license number
Web URL	Social Security Number
Email address	Telephone numbers
Elements of dates: birthdate, admission, discharge	

GDPR	
Name	Physical factor
Location data	Physiological factor
Online identifier	Genetic factor
Cultural factor	Mental factor
ID number	Economic factor
Social identification number	

even removed, from datasets. An example of these data fields is shown in Table I.

5) De-identification Techniques: The Anonymizer can be customized to indicate what data to anonymize, to replace, or keep as is. The type of techniques considered by the Anonymizer was atomic, that is, they can be used in isolation on specific terms of the log streams. The techniques implemented in the Anonymizer (which have been presented in literature before [14]–[16]) are the following:

- *Tokenization.* This technique is a type of pseudonymization in which an attribute's value is replaced by a "token". For example: converting a credit card number 5315 1111 1111 1234 to ****1234.
- *Hashing.* This pseudonymization technique maps data of any size to a code of a fixed length using a hash function. There are different hash options such as MD5 and SHA. Although the hashing cannot be reversed, the original values can be derived if the range of input values is known so they can be replayed to the function [17]. For example, converting a medallion number of value 9Y99 to the hash string 71b9c3f3ee5efb81ca05e9b90c91c88f.
- *Hiding/Suppression/Blackmarking.* This technique involves fully removing the value of an attribute by replacing it with a constant (known as "black marker") so that no value is released at all. As a result, all the information about the attribute is lost and only metrics about the overall dataset can be computed. Typically, this form of anonymization would be used when sensitive attributes are not required for data analysis. For example: transforming medical diagnosis *cancer* to *; or all IP addresses in a log trace to a same IP address 10.11.1.1.
- *Permutation.* It consists of randomly shuffling some values (e.g., the sensitive ones) among records such that some relationships between attributes are deassociated. This technique preserves the distribution and the range of values but it destroys the correlations between entities and values.
- *Shift.* It is a type of noise-additive technique that involves adding a fixed offset ("shift") to a data value in order to reduce the risk of reidentifiability. Since the shift value is fixed, the ordering of the values within an attribute is preserved (e.g., ranking values is still possible). The data can be de-anonymized by the data controller if s/he preserves the shift value. For example, adding 10cm to the height of people.
- *Truncation.* It consists of removing a certain amount of data from the end of a field. This technique hides the true value of the data while still providing some relevant information. For example, converting the IP address 192.168.2.12 to 192.168.0.0 (preserving the first two octets) or the ZIP code 77063 to 77*** (preserving the state and city).
- *Prefix-Preserving.* This is the process of retaining the first n-bits of the original data value of size m while scrambling the remaining (m-n)-bits. This is useful to respect the general structure of an item of data, but not displaying the original data itself.
- *Encryption.* It consists on encoding a value such that only authorized parties are able to access it. Encryption hides completely the value of an attribute, as it loses the semantics of the value and makes the data unreadable. In our work, we used the Java cryptography classes to support the Blowfish algorithm [18].
- *Aggregation.* This technique consists of grouping records of entities together such that they all share the same value in certain attributes. Aggregation prevents singling out an entity in a group. An example can be to make locations of individuals more coarse-grained, so instead of showing a particular place (e.g., oncology clinic), the category is shown (e.g., health facility).

III. EXPERIMENTAL SETUP

We conducted a series of experiments that pursued two objectives: (1) to investigate the level of data protection that reduction and anonymization of logs can offer to avoid exposure of sensitive information; (2) to evaluate the effectiveness of the anonymized logs in task-specific analysis. For this purpose, we developed a prototype in Java implementing the series of techniques discussed in Section II-B. We processed log and event data through our prototype (reduction and anonymization), push the data to a log management system (i.e., EventLog Analyzer [12]) and then conducted anomaly detection analysis to evaluate the utility of the anonymized logs.

As evaluation data, we utilized two datasets, widely used in the literature, which contain traces from web logs and

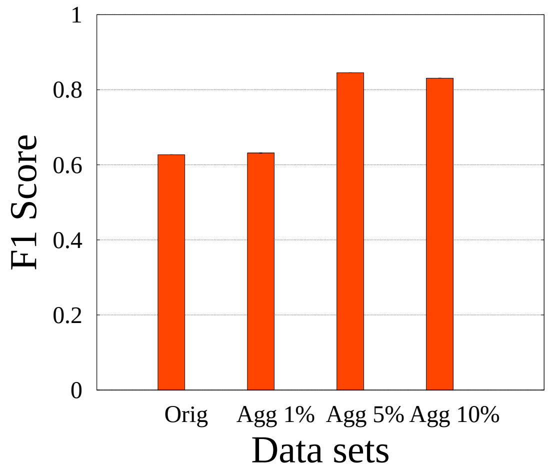

Figure 3. LSTM Accuracy - F1 score

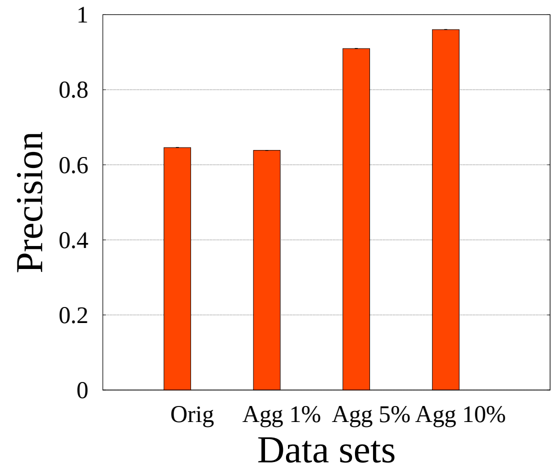

Figure 4. LSTM Accuracy - Precision

application services. Some of the sensitive information in these datasets that a company would be interested in protecting are usernames, IP addresses, port numbers, node names, application names, path URLs, user types, passwords, email addresses, etc. The datasets are: *(1) NASA-HTTP dataset.* It contains traces of HTTP requests performed over two months to the NASA Space Center server in Florida [19]. The logs are structured as one line per request with the following columns: host making the request, timestamp, request given in quotes, HTTP reply code, and bytes in the reply. *(2) Yahoo dataset.* It is used to benchmark anomaly detection algorithms [20] and consists of time-series (with tagged anomaly points) representing the metrics of various Yahoo services.

IV. Experimental Results

Our discussion of experimental results is presented with respect to the type of analysis performed over the log streams, evaluating the impact of SafeLog in those tasks. In these experiments, we adopted a privacy-by-default approach in which only the fields required for the analysis were mainly preserved.

A. Anomaly Detection Analysis

Firstly, we studied the impact of anonymization for anomaly detection in the usage of 67 Yahoo services. In this context, an anomaly is defined as any event that does not conform to the pattern suggested by the model [21]. To this end, we ran two widely popular machine learning algorithms (i.e., Autoregressive Integrated Moving Average (ARIMA) modeland Long Short Term Memory (LSTM) networks [22]) against the original and the anonymized logs. Then, we compare the performance of the algorithms in terms of three well-known accuracy measures to measure the impact of our approach: Precision, recall, and F1-score [23]. In all cases, 1 is the highest value and 0 the lowest.

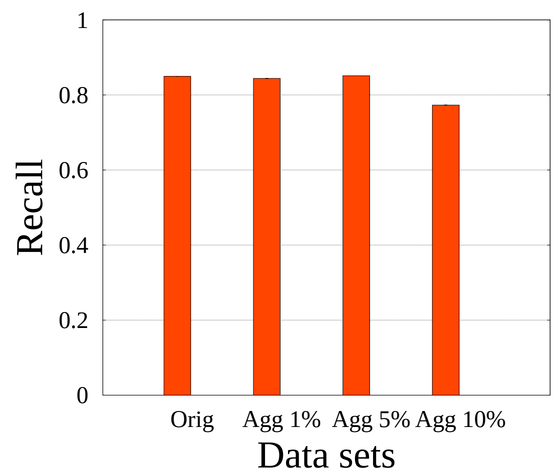

Figure 5. LSTM Accuracy - Recall

In general, the results showed that aggregating the data (before feeding it to the algorithms) benefits the identification of anomalies. To illustrate this, Fig. 3 shows the average F1 score achieved by each type of logs. It can be noticed how conducting a minimum aggregation (i.e., grouping the numeric data in clusters representing 1% of the dataset's cardinality) practically achieved the same F1 score than the original log (which is the baseline in this analysis). Then, higher levels of aggregation (e.g., 5 or 10%) achieved a considerably higher F1 score (i.e., more than 20% improvement). This positive impact was the result of reducing the diversity of the values fed to the algorithms, which consequently helped to produce smoother models that were more effective at identifying anomalies.

Additionally, the same trend was experienced by the precision metric (shown in Fig. 4). Regarding the recall (depicted in Fig. 5), the behavior was slightly different: A similar level of accuracy (compared to the original log)

```
pm032-04.dialip.mich.net - - [11/Jul/1995:17:55:58 -0400] "GET / " 200 7062
poppy.hensa.ac.uk - - [11/Jul/1995:17:55:58 -0400] "GET /shuttle/missions/sts-67/images/KSC-95EC-0397.jpg " 200 125249
sgigate.sgi.com - - [11/Jul/1995:17:56:01 -0400] "GET /shuttle/technology/sts-newsref/sts_asm.html " 200 71656
pm032-04.dialip.mich.net - - [11/Jul/1995:17:56:01 -0400] "GET /images/ksclogo-medium.gif " 304 0
stratacom.strata.com - - [11/Jul/1995:17:56:02 -0400] "GET /images/NASA-logosmall.gif " 304 0
sgigate.sgi.com - - [11/Jul/1995:17:56:03 -0400] "GET /shuttle/technology/images/srb_mod_compare_6-small.gif " 200 28219
sgigate.sgi.com - - [11/Jul/1995:17:56:03 -0400] "GET /shuttle/technology/images/srb_mod_compare_1-small.gif " 200 36902
blazemonger.pc.cc.cmu.edu - - [11/Jul/1995:17:56:06 -0400] "GET /htbin/cdt_clock.pl HTTP/1.0From:
<berend@blazemonger.pc.cc.cmu.edu>" 200 751
wc40.writer.yorku.ca - - [11/Jul/1995:17:56:06 -0400] "GET /shuttle/missions/sts-36/sts-36-patch-small.gif " 200 10923
134.241.78.27 - - [11/Jul/1995:17:56:07 -0400] "GET /shuttle/missions/sts-71/images/KSC-95EC-0912.jpg " 200 57344
```

Figure 6. Sample of NASA web logs

was achieved by the 1 and 5% aggregated datasets. Then, the accuracy slightly decreased (by an average of 7%) in the 10% aggregation. This was caused by the Yahoo logs being relatively unbalanced (in terms of actual anomalies), so when one is missed it has a substantial effect on the score. It should also be noted that because the Yahoo logs represent a wide range of different services' behaviours, the standard deviations were relatively high for the original log data (ranging between 20% and 39%). However, the standard deviation considerably decreased after aggregation took place, which was another positive indicator that the overall accuracy was improved (e.g., it only ranged between 9% and 23% for the 10% aggregation). Finally, despite some (expected) differences in performance, similar trends were observed in the Arima results for all relevant metrics.

B. Users Behavior Analysis

Here, we used the NASA dataset to extract common statistics derived from web logs. In particular, we present next two mining goals typically carried out by a web analyst, discussing if the goals were still fulfilled after using our data protection approach.

Goal 1: To identify the most visited pages from the website and the most popular visiting times. To this end, the analyst requires to obtain a list of the URLs visited (in order to perform a count of the number of times they were visited) and the timestamp of such visits (to identify the busiest times). As seen in Fig. 6, the original logs overexpose information such as the IP address and email of visitors, considered personal data under GDPR, and which is not required to perform this task. Hence, the IPs were truncated such that the 8 least significant bits were replaced with 0. As per the emails, these were hidden such that only the domain name was preserved. These techniques allowed to conduct the analysis without any impact on their accuracy (as shown in Figs. 7 and 8).

Goal 2: To identify areas of crawl waste [24] in a website to fix them. That is, companies intent to optimize the time spent by search engines crawling its website to make it discoverable by potential visitors. To this end, an analyst inspects the logs to identify low value-add URLs (e.g., pages

Request	Count	Percentage
get /images/nasa-logosmall.gif	182830	4.93%
get /images/ksc-logosmall.gif	151425	4.08%
get /images/mosaic-logosmall.gif	92970	2.5%
get /images/usa-logosmall.gif	92290	2.49%
get /images/world-logosmall.gif	91320	2.46%
get /images/ksclogo-medium.gif	91135	2.46%
get /shuttle/countdown/	82133	2.21%
get /images/launch-logo.gif	65167	1.76%
get /shuttle/countdown/count.gif	65077	1.75%
get /ksc.html	62236	1.68%

Figure 7. Most Visited Pages

Figure 8. Top Visiting Dates

with high load time or redirecting links). Some specific analysis actions would involve recognizing URLs with session IDs, faceted navigation, infinite spaces, duplicates or containing parameters (i.e., URLs with a question mark). In our analysis, we focused on inspecting the status codes of the HTTP requests to identify redirections (304, 302), client errors (404), and server errors (501). Hence, we filtered out

Status Code	Transfered Bytes	Request	Time
304	0	GET /history/apollo/images/apollo-logo1.gif HTTP/1.0	28 Jul 1995, 13:31:53
304	0	GET /images/KSC-logosmall.gif HTTP/1.0	28 Jul 1995, 13:31:53
304	0	GET /history/apollo/images/apollo-logo1.gif	28 Jul 1995, 13:31:53
304	0	GET /images/KSC-logosmall.gif	28 Jul 1995, 13:31:53
302	85	GET /cgi-bin/imagemap/countdown70?277,289 HTTP/1.0	28 Jul 1995, 13:29:53
302	85	GET /cgi-bin/imagemap/countdown70?277,289	28 Jul 1995, 13:29:53
302	85	GET /cgi-bin/imagemap/countdown70?277,289	28 Jul 1995, 13:29:53
302	102	GET /cgi-bin/imagemap/countdown70?11,351 HTTP/1.0	28 Jul 1995, 13:29:37
404	-	GET /images/lf-logo.gif	28 Jul 1995, 13:32:11
404	-	GET /images/crawlerway-logo.gif HTTP/1.0	28 Jul 1995, 13:31:10
501	-	POST /shuttle/missions/sts-69/mission-sts-69.html	27 Jul 1995, 11:44:31
501	-	POST /images/getstats.gif HTTP/1.0	26 Jul 1995, 09:27:10

Figure 9. Budget Crawl Sources

all data (i.e., data minimization by default) except the one required to perform this analysis. As depicted in Fig. 9, we preserved the required data, involving status codes, the transferred bytes (as it can impact the load time of a page), the requested resource (as broken resources are a dead end for search engines), and the time (in order to know when the event occurred).

In conclusion, this experiment proved that it was possible to conduct the desired behaviour analysis (after using our approach) to better protect the data privacy. An additional benefit obtained from using the approach was a considerable reduction in the amount of data sent to the log management service (i.e., 65%). This can help to reduce the costs of using this type of cloud-based services (which typically charge per usage).

V. Related Work

Although sharing log data with third parties brings several benefits for a company, it also introduces a threat to users' privacy. For this reason, several research efforts have been concentrated on developing techniques and tools that can assist companies in anonymizing this type of data. FLAIM [15], TCPDPRIV [25], and CryptoPAn [26] are tools for sanitizing network traces mainly aimed at system administrators or security engineers. Other relevant tools focused on structured data anonymization include research prototypes and industrial-oriented tools such as TIAMAT [27], Cornell Anonymization Toolkit [28], UTD Anonymization Toolbox [29], ARX [30], and SECRETA [31]. In the area of statistical disclosure control (SDC), there are two widely-known tools that are commonly used by statistical agencies in the European Union [32]: μ-Argus and sdcMicro. μ-Argus [33] is an open-source software that implements techniques like microaggregation, PRAM, rank swapping to sanitize microdata; while sdcMicro [34] is an open-source package for the R statistical software which includes all the methods from μ-Argus plus a set of additional new ones.

Unlike traditional sanitization programs that work on low-level systems' data, our approach works on a higher layer that can be easily integrated with cloud-based log management services. Moreover, we integrate a data minimization functionality. Our evaluation is highly oriented on demonstrating the utility of the data after data minimization and anonymization, whereas most of the related work is mainly evaluated with respect to privacy concerns only without exploring task-specific applications [35]. Most of the existing work has evaluated the utility of anonymized network traces for detecting security threats [36], [37]. Some of those works utilize anomaly detection as a technique for detecting scans and denial of service attacks. In our work, the anomaly detection evaluation is performed in a broader perspective as it can be used for different types of services.

VI. Conclusions And Future Work

In this paper, we presented SafeLog, which is an approach to provide data protection in log data that is outsourced to log management and analytics systems in the cloud. Additionally, we operationalize the principles of data confidentiality and data minimization through anonymization and filtering, respectively, to facilitate compliance with privacy laws (e.g., GDPR) when demonstrating accountability. Our purpose is to reduce the risk of data misuse and breaches by limiting the exposure of sensitive and personal data, while still allowing companies to obtain insights from analyzing the log streams in the cloud (e.g., identifying threats, anomalies, or understanding users' and systems' behaviors).

Our results demonstrated that various benefits can be achieved from our approach: the processed logs (reduced and anonymized) were still useful to conduct two analysis tasks typically performed in an industrial web environment (i.e., identifying user behaviours and detecting anomalies in services); the financial costs of using cloud-based log management services is decreased, as our approach removes data which is not essential for the desired task (hence reducing the amount of data sent to the cloud); the use of atomic anonymization operations allows applying a more fine-grained anonymization than what is commonly applied, which increased data utility; the definition of privacy patterns allow the easy identification and transformation (or removal) of data that needs to be protected, which can be customized according to data protection and privacy laws. Finally, if personal data becomes strongly and irreversible anonymized, it does not longer fall under the scope of the most recent privacy laws, such as GDPR.

As future work, we plan to extend the capabilities of SafeLog by supporting other types of anonymization techniques. We will also strengthen the evaluation of the approach (e.g., by diversifying the tested datasets and broadening the set of evaluated analysis tasks). Finally, we plan to enhance the quality of our prototype (e.g., by developing a Graphical User Interface) in order to release it as a publicly-available tool.

References

[1] International Association of Privacy Professionals, "The General Data Protection Regulation Matchup Series," https://iapp.org/resources/article/the-general-data-protection-regulation-matchup-series/, Last accessed: 2019-08-04.

[2] "GDPR chapters and recitals," last accessed: 2019-08-04. [Online]. Available: https://gdpr-info.eu/

[3] V. Ayala-Rivera and L. Pasquale, "The grace period has ended: An approach to operationalize gdpr requirements," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*. IEEE, 2018, pp. 136–146.

[4] J. Domingo-Ferrer and J. Soria-Comas, "Anonymization in the time of big data," in *International Conference on Privacy in Statistical Databases*. Springer, 2016, pp. 57–68.

[5] "Plaintext Passwords in Twitter Internal Logs," last accessed: 2019-08-04. [Online]. Available: https://www.bleepingcomputer.com/news/security/twitter-admits-recording-plaintext-passwords-in-internal-logs-just-like-github/

[6] "Plaintext Passwords in GitHub Logs," last accessed: 2019-08-04. [Online]. Available: https://www.bleepingcomputer.com/news/security/github-accidentally-recorded-some-plaintext-passwords-in-its-internal-logs/

[7] "Third-party vendor risk and breach likelihood," last accessed: 2019-08-04. [Online]. Available: https://iapp.org/news/a/surprising-stats-on-third-party-vendor-risk-and-breach-likelihood/

[8] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, L. Murphy *et al.*, "A systematic comparison and evaluation of k-anonymization algorithms for practitioners," *Transactions on data privacy*, vol. 7, no. 3, pp. 337–370, 2014.

[9] D. Goodin, "Poorly anonymized logs reveal nyc cab drivers detailed whereabouts," *Ars Technica*, 2014.

[10] K. El Emam, *Data anonymization practices in clinical research: a descriptive study*. CHEO Research Institute, 2011.

[11] "DataDog," last accessed: 2019-08-04. [Online]. Available: https://docs.datadoghq.com/

[12] "EventLog Analyzer," last accessed: 2019-08-04. [Online]. Available: https://www.manageengine.com/products/eventlog/

[13] "LogEntries by Rapid7," last accessed: 2019-08-04. [Online]. Available: https://logentries.com/

[14] "Opinion 05/2014 on Anonymisation Techniques," last accessed: 2019-08-04. [Online]. Available: http://ec.europa.eu/newsroom/article29/news-overview.cfm

[15] A. J. Slagell, K. Lakkaraju, and K. Luo, "Flaim: A multi-level anonymization framework for computer and network logs." in *LISA*, vol. 6, 2006, pp. 3–8.

[16] V. Ayala-Rivera, D. Nowak, and P. McDonagh, "Protecting organizational data confidentiality in the cloud using a high-performance aonymization engine," 2013.

[17] D. Goodin, "Poorly anonymized logs reveal NYC cab," last accessed: 2019-08-04. [Online]. Available: https://arstechnica.com/?post_type=post&p=492413

[18] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in *Int. Workshop on Fast Software Encryption*. Springer, 1993, pp. 191–204.

[19] "NASA-HTTP dataset," last accessed: 2019-08-04. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

[20] "Yahoo Anomaly Detection Dataset," last accessed: 2019-08-04. [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[21] A. Gulenko, M. Wallschlager, F. Schmidt, O. Kao, and F. Liu, "Evaluating machine learning algorithms for anomaly detection in clouds," *IEEE International Conference on Big Data*, pp. 2716–2721, 2016.

[22] S. Brady, D. Magoni, J. Murphy, H. Assem, and A. O. Portillo-Dominguez, "Analysis of machine learning techniques for anomaly detection in the internet of things," in *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE, 2018, pp. 1–6.

[23] F. Almaguer-Angeles, J. Murphy, L. Murphy, and A. O. Portillo-Dominguez, "Choosing machine learning algorithms for anomaly detection in smart building iot scenarios," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. IEEE, 2019, pp. 491–495.

[24] "Crawl Budget," last accessed: 2019-08-04. [Online]. Available: https://www.contentkingapp.com/academy/crawl-budget/

[25] "TCPDPRIV," last accessed: 2019-08-04. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html

[26] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[27] C. Dai, G. Ghinita, E. Bertino, J.-W. Byun, and N. Li, "Tiamat: a tool for interactive analysis of microdata anonymization techniques," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1618–1621, 2009.

[28] X. Xiao, G. Wang, and J. Gehrke, "Interactive anonymization of sensitive data," in *ACM SIGMOD*. ACM, 2009, pp. 1051–1054.

[29] "UTD ToolBox." http://cs.utdallas.edu/dspl/cgi-bin/toolbox/, last accessed: 2019-08-04.

[30] F. Prasser and F. Kohlmayer, "Putting statistical disclosure control into practice: The arx data anonymization tool," in *Medical Data Privacy Handbook*. Springer, 2015, pp. 111–148.

[31] G. Poulis and A. Gkoulalas-Divanis, "SECRETA: A System for Evaluating and Comparing RElational and Transaction Anonymization algorithms." *International Conference on Extending Database Technology*, pp. 620–623, 2014.

[32] T. Benschop, C. Machingauta, and M. Welch, "Statistical Disclosure Control for Microdata: A Practice Guide," International Household Survey Network and the World Bank, Tech. Rep. July, 2016.

[33] A. Hundepool, P.-P. de Wolf, J. Bakker, A. Reedijk, L. Franconi, S. Polettini, A. Capobianchi, and J. Domingo, "μ-ARGUS version 5.1 User's Manual," Statistics Netherlands, Tech. Rep. 20, 2014.

[34] "sdcMicro: Statistical Disclosure Control Methods for Anonymization of Microdata and Risk Estimation," https://cran.r-project.org/web/packages/sdcMicro/, last accessed: 2019-08-04.

[35] S. E. Coull, C. V. Wright, A. D. Keromytis, F. Monrose, and M. K. Reiter, "Taming the devil: Techniques for evaluating anonymized network data." in *NDSS*, 2008.

[36] K. Lakkaraju and A. Slagell, "Evaluating the utility of anonymized network traces for intrusion detection," in *4th Int. Conf. on Security and privacy in communication networks*. ACM, 2008.

[37] M. Burkhart, D. Brauckhoff, and M. May, "On the utility of anonymized flow traces for anomaly detection," *arXiv preprint arXiv:0810.1655*, 2008.