Data Cleaning Technique for Security Big Data Ecosystem

Diana Martínez-Mosquera[1] and Sergio Luján-Mora[2]
[1]*Departamento de Electrónica, Telecomunicaciones y Redes de Información, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito, Ecuador*
[2]*Department of Software and Computing Systems, University of Alicante, Spain*

Keywords: Data, Cleaning, Big Data, Security, Ecosystem.

Abstract: The information networks growth have given rise to an ever-multiplying number of security threats; it is the reason some information networks currently have incorporated a Computer Security Incident Response Team (CSIRT) responsible for monitoring all the events that occur in the network, especially those affecting data security. We can imagine thousands or even millions of events occurring every day and handling such amount of information requires a robust infrastructure. Commercially, there are many available solutions to process this kind of information, however, they are either expensive, or cannot cope with such volume. Furthermore, and most importantly, security information is by nature confidential and sensitive thus, companies should opt to process it internally. Taking as case study a university's CSIRT responsible for 10,000 users, we propose a security Big Data ecosystem to process a high data volume and guarantee the confidentiality. It was noted during implementation that one of the first challenges was the cleaning phase after data extraction, where it was observed that some data could be safely ignored without affecting result's quality, and thus reducing storage size requirements. For this cleaning phase, we propose an intuitive technique and a comparative proposal based on the Fellegi-Sunter theory.

1 INTRODUCTION

Every minute, thousands of activity events are occurring and being registered into network equipment. Monitoring and analyzing this digital information is the important task performed by the Computer Security Incident Response Team (CSIRT), because they provide proactive and reactive support to vulnerabilities and intrusions. However, they are limited to a small time-frame, usually a few days and the ever-overwhelming growth in information size and complexity; due to this drawback, the CSIRT cannot analyze historical security events. Currently, the CSIRT's main issues are storage space constraints, code maintenance, confidential and sensitive information processed by third parties, cost associated with current solution, etc.

Security data can be considered as Big Data, since it complies with five Vs, the defining properties of Big Data (Arputhamary and Arockiam, 2015):

1. Volume, for the high rate of produced information when register all network events.
2. Velocity, since data need to be processed and analyzed as fast as possible in order to detect, prevent and react to possible threats.
3. Variety, event logs are classified as semi-structured data (Khalifa et al., 2016), since they generally contain metadata to describe their structure.
4. Veracity, security information and logs proceed from trusted network elements.
5. Value, since it can alert the security team to threats that can happen, or may be happening, in the network.

In this paper, we propose a Big Data ecosystem for security data based on an open source framework. This choice is due to commercial solutions being expensive, may not be adaptable to company's requirements or peculiarities and code is not available for analysis which may present a confidence liability. The proposed ecosystem allows processing a high data volume, customizing the code according specific requirements, keeping the confidentiality and saving costs (Cárdenas et al., 2013). During the initial implementation phase, it can be noticed that one of the main process is data Extract, Transform and Load

Martínez-Mosquera, D. and Luján-Mora, S.
Data Cleaning Technique for Security Big Data Ecosystem.
DOI: 10.5220/0006360603800385
In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security (IoTBDS 2017)*, pages 380-385
ISBN: 978-989-758-245-5

(ETL). This is confirmed by Intel Corporation, which mentions that data integration stands for 80% of development effort in Big Data projects (Intel, 2013). Extract task gathers data from external sources; Transform task employs a variety of software tools and custom programs to manipulate collected data, cleaning is also addressed on this task to remove unnecessary data; finally, Load task loads the data to permanent storage.

The data integration process into the selected framework has a slight variation and first there is Extract, then Load and finally Transform (ELT). However, and after observing raw data (possible due to log information being plain text files) we concluded that some information was irrelevant or redundant, for instance, the producer name, domain names that can be resolved by IP addresses, etc. Thus, and with a simple test by manually removing the irrelevant fields, the size of the information could be reduced before the Load stage and more data could be stored without quality loss.

This background suggested the development of an intuitive algorithm for removing useless fields after data extraction process. The attained results through a developed script were satisfactory but important scalability and flexibility issues need to be resolved, due to human intervention requirement in changing and maintaining said script.

Several data cleaning techniques and tools have been developed, but the major part of them target data deduplication while our requirements were the useless data cleaning. In this article, we limit to present a Security Big Data ecosystem testbed and an intuitive data cleaning technique with some results and improvement proposals, for instance, increase data retention by at least 25%. This solution has been tested with data provided by responsible for 10,000 university users.

One of the main challenges in this research is adapting data cleaning techniques to security data in Big Data ecosystems to overcome storage space constraints and we intend to solve the following questions:

- Is it possible to define a Big Data security ecosystem?
- Is it possible to apply data cleaning techniques to security data to reduce storage space?

2 STATE OF THE ART

Commercially, there are several available solutions to process Big Data, however, as we already mentioned before, they are expensive and the inclusion of a third party can introduce security and confidentiality liabilities that are unacceptable to some companies. A Big Data ecosystem must be based on the following six pillars: Storage, Processing, Orchestration, Assistance, Interfacing, and Deployment (Khalifa et al., 2016). The authors mention that solutions must be scalable and provide an extensible architecture so that new functionalities can be plugged in with minimal modifications to the whole framework. Furthermore, the need for an abstraction layer is highlighted in order to augment multi-structured data processing capabilities.

Apache Hadoop is a free licensed distributed framework that allows working with thousands of independent computers to process Big Data (Bhandare, Barua and Nagare, 2013). Stratosphere is another system comparable to Apache Hadoop, which, according to the authors, its main advantage is the existence of a pipeline, which improves execution performance and optimization; although being released in 2013 it has not been as widely used as Hadoop (Alexandrov et al., 2014).

To support the data cleaning process, statistical outlier detection, pattern matching, clustering and data mining techniques are some of the available techniques to data cleaning tasks. However, a survey (Maletic and Marcus, 2009) evidences that customized process for data cleaning are use in real implementations. Thus, there is a need for building high quality tools.

In a work about data cleaning methodologies for data warehouse, data quality is assured but there is not a clear path on how these techniques can be adapted to our interests (Brizan et al., 2006). BigDansing technique is targeted to Big Data (Khayyat et al., 2015), but, similarly to other techniques, its main purpose is to remove inconsistencies on stored data. MaSSEETL, an open-source tool, is used over structured data and works on transforming stored data when our research focus on extraction and cleaning (Gill and Singh, 2014).

Data cleaning is often an iterative process adapted to specific task's requirements; a survey to data analysts and industry's infrastructure engineers shows that data cleaning is still an expensive and time-consuming activity. Despite community's research and the development of new algorithms, current methodology still requires the existence of human-in-the-loop stages and that both data and result be evaluated repeatedly, thus several challenges are faced during design and implementation. Data cleaning is also a complex process that involves extraction, schema/ontology matching, value imputation, de-duplication, etc.; additionally, each of these tasks encapsulates several specialized algorithms such as machine learning, clustering or rule based procedures. In this research, it is proposed the idea of making an easy and fully automated data cleaning process (Krishnan, Haas, Franklin and Wu, 2016).

Data cleaning methods allow duplicated data to be found within files or group of files. It can be explained through Fellegi-Sunter mathematical model, which does not need trained data but do requires optimal parameters estimation (Winkler, 2003).

Advanced methods like Bridging Files use a master file maintained by a trustworthy unit that contains exact and updated information, so that comparisons can be made only against that file. For structured information, it is considered a good method; however, in the security context it may not be adequate, as it would need to analyze IP addresses, ports, every entities equipment, etc. Having a unit that will hold this information is unlikely and said unit would introduce human intervention in the loop (Winkler, 2003).

Big Match Technology has been applied in de-duplication between files A and B, where B contains indexes that allow higher information volumes to be processed with less resource. Although, it has been applied in scenarios related to people and cities information, no results are available in security information field. There are related methods briefly explained like:

1. Preprocessing and standard methods to identify sub-fields.
2. Advanced chain comparators.
3. Analytic link methods.

The three methods focus on information de-duplication, but we are intending to target them at finding useful data while eliminating the unnecessary (Winkler, 2003).

Our research was unable to reach a suitable ecosystem for security Big Data; therefore, a specific ecosystem was built and the first steps are the data extraction and data loading.

3 METHOD

3.1 Security Big Data Ecosystem

Among the different available frameworks to build Big Data ecosystems, we selected Apache Hadoop. Apache Hadoop, being an open-source framework, provides transparency and adaptability while also being a proven cost-effective solution in several Big Data scenarios. Apache Hadoop is included in several products like Microsoft Azure HDInsight, Cloudera, among others (Nehe, 2016).

Table 1: Proposed Ecosystem.

	TOOL	**ADVANTAGE**
STORAGE	High Distribution File System	Store any data format High data scalable
PROCESSING	Map Reduce	Fault tolerant No human intervention
ORCHESTRATION	Yet Another Resource Negotiator (YARN)	Increase resource usage
ASSISTANCE	Help Pages	Easy to develop
INTERFACE	Pig	High flexible
DEPLOYMENT	VBlock	Faster to setup

Based on the six mentioned fundamental pillars Storage, Processing, Orchestration, Assistance, Interfacing and Deployment, our proposed Security Big Data Ecosystem is presented on Table 1, it describes the used tools for each pillar and the main advantages (Khalifa et al., 2016).

Figure 1 shows the logic topology of the ecosystem, the source of the security log files transfers the data to the Hadoop Distributed File System (HDFS), YARN defines the abstraction layer between HDFS and Map Reduce, the paradigm to transform the data set in a pair key/value, and Pig is used for analyzing large data sets.

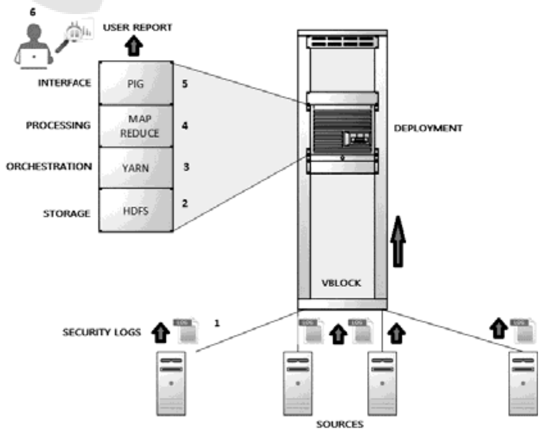

Figure 1: Security Big Data Ecosystem.

3.2 ETL Process

In order to implement a Big Data ecosystem, the first steps are the data extraction and data loading. Here, the ETL stages come to mind, Figure 2 describes the three applied tasks in a traditional data warehousing process (Arputhamary and Arockiam, 2015).

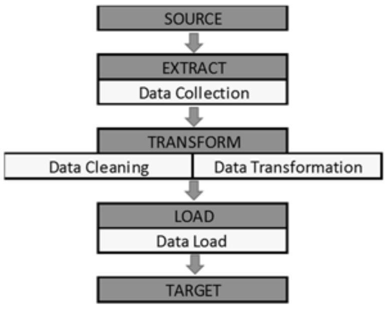

Figure 2: ETL Process.

In Hadoop, ETL process becomes Extract, Load and Transform (ELT) targeting processing time reduction; however, in practice, during data loading step it was clear the existence of useless and redundant fields and transformation process improves if data is cleaned first, thus, we propose: Extract, Cleaning, Load and Transform (ECLT). Extract task will collect log files from the sources, Cleaning task, will perform the data cleaning, Load will store the data into the HDFS and Map Reduce will do Transform step.

For testing the Cleaning task, security log files from a firewall were used. This firewall, belonging to university's CSIRT, produces around 30,000 rows of log per hour.

3.2.1 Intuitive Proposal for Data Cleaning

For most systems, data cleaning process ignores data for certain hours or equipment, thus, it is possible to reduce the size of the data to be analyzed in a Big Data ecosystem. However, security data cannot be ignored since any time an attack can be present in any equipment. Therefore, we proposed an intuitive technique for security data cleaning in an ECLT process. This intuitive technique requires human intervention to analyse the relevant security data.

For instance, a sample of security log contains the following header fields: Product Name, Severity, Event Name, Start Time, Source Country, Source, Destination Country, Destination, Service, Attack Name, Attack Information, User, Verdict, Total connections, Event ID. Security data analyst found the fields that can be discarded during data cleaning process: Product Name, Source Country, Destination Country, User, Verdict and Event ID.

Through a bash script, it was possible to clean the security log file, for example, the main code line:

```
cat LOG_BEFORE_CLEANING | cut -d "," -f
x,y,z > LOG_AFTER_CLEANING
```

Where x, y, z variables correspond to relevant information to be preserved and we can differentiate every one of them by the delimiter comma “,” since as we mentioned before security log files are semi structured data.

We tested the script for 32 security log samples, 10 minutes of data (around 5,000 lines), 100 minutes of data (around 50,000 lines), 24 hours, a day of data (around 720,000 lines), two, three, four until thirty days. Some attained results are presented on Table 2, there we can see for 10 minutes log file the size decreased from 1.2 MB to 0.87 MB, therefore 27.5 % less than the original file, for 100 minutes log file the size decreased 29.2%, for a day log file the size decreased 28.9% and so on.

As it can be observed in Figure 3, size reductions between 25% and 30% can be achieved with this mechanism and the results keep constant for the 32 tests, the attained average is 28.9%. With it, storage costs can be reduced or data retention capability increased by at least 25%. Comparing our attained results with other researches, the main difference is probably the nature of the data, since they propose to discard data at row level (Aye, 2011), and we consider important all the security log rows; for that reason, our data cleaning proposal is based only in a vertical dimension.

Table 2: Intuitive Method Results.

ORIGINAL SIZE [MB]	**TIME [days]**	**SIZE AFTER DATA CLEANING [MB]**	**REDUCTION [%]**
1.2	0.007	0.87	27.5
12	0.07	8.5	29.2
173	1	123	28.9
345	2	245	29.0
517	3	367	29.0
690	4	490	29.0
862	5	612	29.0
1100	6	734	33.3
1200	7	856	28.7
1400	8	979	30.1
1600	9	1100	31.3
1700	10	1200	29.4

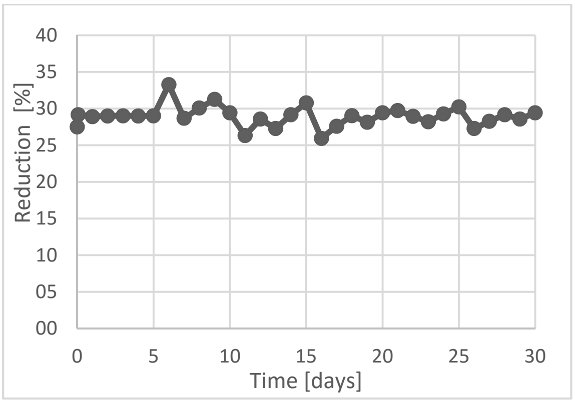

Figure 3: Time vs Reduction.

This intuitive solution is easily implementable and gives good results, however, this proposal does not fully comply some main characteristics to be considered as a suitable solution (Arputhamary and Arockiam, 2015): Reliability, Maintainability, Freshness, Recoverability, Scalability, Availability, Traceability, Auditability and Maintenance. For instance, Scalability since the code lines for data cleaning must be created in house according the company's requirements, automatic Recoverability due to the need of the human intervention to restore the data cleaning script. Hence, we need to avoid the human intervention in the process (Krishnan et al., 2016).

With this intent, some automatic data cleaning techniques were analysed in security log scenarios, but we were unable to find a suitable procedure that would allow us to reach the intended goal. For this reasons, the paper also presents a comparative proposal based on a mathematic model to surpass the mentioned constrains. The next section describes the details of this comparative proposal.

3.2.2 Comparative Proposal based on Fellegi-Sunter Theory

Data cleaning rules vary from organization to organization according to their requirements, so we are working in the development of a technique for automatically creating these rules. For that, we need to identify the useful fields to maintain and useless ones to delete them from original log files. Fellegi-Sunter model was selected since it describes a mathematical theory for record linkage without the need for data training. They offer a framework for solutions focused on record recognition from two files that can represent people, objects or events, the last one being our point of interest (Fellegi and Sunter, 1969).

Our main goal is to avoid the human intervention in the data cleaning process, thus, the irrelevant data to be removed could be identified comparing two files, the original security log file and the final user security report. This last one will have the useful data, for instance, for horizontal, vertical and box scanning reports, the useful fields are source and target IP addresses, ports, domains and time.

Fellegi and Sunter defined three sets of elements, A1, A2 and A3, where A1 corresponds to a match between two files; A3 corresponds to a non-match between two files and A2 to a possible match between two files (Fellegi and Sunter, 1969). As A, we will denote the set of header fields in the final user security report. As B, we will denote the set of header fields in the original security log file (Winkler, 1988).

According to Fellegi and Sunter notation, A1 will correspond to matched elements between A and B, therefore, A1 will contain the set of fields that must be adding to the variables x, y, z into the data cleaning script. If the algorithm cannot decide if the element belongs to A1 or A3, the Fellegi-Sunter corollaries will be used to determine the associated error level (Fellegi and Sunter, 1969). We are working to implement this comparative technique and present the obtained results; the main challenge is to compare the original log file and the final user report since they have no similar header field names.

4 CONCLUSIONS

In this paper, we have presented a proposed security Big Data ecosystem based on six fundamental pillars Storage, Processing, Orchestration, Assistance, Interfacing and Deployment. Moreover, we have presented the initial findings from a data cleaning technique in a Security Big Data Ecosystem designed for a university with 10,000 users.

The data cleaning technique for security log files is an intuitive one but, apart from the acceptable results, size reductions between 25% and 30%, it is not scalable since the code needs to change every time we need to stablish new rules or adapt new scenarios. With the use of a mathematical theory, defined by Fellegi and Sunter, we propose to improve the technique to automate data cleaning process, gathering the useful fields from the user reports and thus avoid the human in the loop problem.

Future work is still needed to implement the developed comparative proposal between user final report and security log file. Our solution is focused on security data and the use of Big Data techniques on this context, two topics that had not been deeply studied together and we hope to take advantage of the opportunities available in these systems.

ACKNOWLEDGEMENTS

We thank to the EPN'S CSIRT for the facilities to test this data cleaning technique.

REFERENCES

Alexandrov, Bergmann R., Freytag S., Hueske F., Heise A., Kao O., Leich M., Leser U., Markl V., Naumann F., Peters M., Rheinländer A., Sax M., Schelter S., Höger M., Tzoumas K., Warneke D., 2014. The Stratosphere Platform for Big Data Analytics. In *VLDB Journal*, vol. 23, no 6, pp. 939–964.

Arputhamary B., Arockiam L., 2015. Data Integration in Big Data Environment. In *Bonfring International Journal of Data Mining*, vol. 5, no 1, pp. 1-5.

Aye, T. T., 2011. Web log cleaning for mining of web usage patterns. In *Third International Conference on Computer Research and Development*, vol. 2, pp. 490-494.

Bhandare M., Barua K., Nagare V., 2013. Generic Log Analyzer Using Hadoop Map Reduce Framework. In *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no 9, pp. 603-607.

Brizan, Guy D., Tansel, Uz A., 2006. Survey of Entity Resolution and Record Linkage Methodologies. In *Communications of the International Information Management Association*, vol. 6, no 3, pp. 41-50.

Cárdenas A., Manadhata P., Rajan S., 2015. Big Data Analytics for Security. In *IEEE Security & Privacy*, vol. 11, no 6, pp. 74-76.

Fellegi, I. P., Sunter A. B., 1969. A Theory for Record Linkage. In *Journal of the American Statistical Association*, vol. 64, no 328, pp. 1183-1210.

Gill R., Singh J., 2014. An Open Source ETL Tool-Medium and Small Scale Enterprise ETL (MaSSEETL). In *International Journal of Computer Applications*, vol. 108, no 4, pp. 15-22.

Intel white paper, 2013. Extract, Transform, and Load Big Data with Apache Hadoop. Available at: https://software.intel.com/sites/default/files/article/402274/etl-big-data-with-hadoop.pdf. Intel Corporation.

Khalifa S., Elshater Y.,Sundaravarathan K., Bhat A., Martin P., Iman F., Rope D., McRoberts M., Statchuk C., 2016. The Six Pillars for Building Big Data Analytics Ecosystems. In *ACM Computing Surveys,* vol. 49, no 2, pp. 33:1-33:35.

Khayyat Z., Ilyas I. F., Jindal A., Madden S., Ouzzani M., Papotti P., Yin S., 2015. Bigdansing: A System for Big Data Cleansing. In *ACM SIGMOD International Conference on Management of Data*, pp. 1215-1230.

Krishnan S., Haas D., Franklin M., Wu E., 2016. Towards Reliable Interactive Data Cleaning: A User Survey and Recommendations. In A*CM SIGMOD/PODS Conference Workshop on Human In the Loop Data Analytics*, p. 9.

Maletic, J. I., Marcus, A., 2009. Data cleansing: A prelude to knowledge discovery. In *Data Mining and Knowledge Discovery Handbook.* Ed. by Maimon, O., Rokach, L. US :Springer, pp. 19-32.

Nehe M., 2016. Malware and Log file Analysis Using Hadoop and Map Reduce. In *International Journal of Engineering Development and Research*, vol. 4, no 2, pp. 529-533.

Winkler W. E., 1988. Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, vol. 667, pp. 671.

Winkler W. E., 2003. Data Cleaning Methods. In *Proceedings of the ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*.