Security Enhancement through Effective Encrypted Communication using ELK

Marco Sánchez
Escuela Politécnica Nacional
Quito, Ecuador
marco.sanchez01@epn.edu.ec

Luis Urquiza
Escuela Politécnica Nacional
Quito, Ecuador
luis.urquiza@epn.edu.ec

ABSTRACT
Big Data has become an important and essential tool for data analysis and decision making due to its fast evolution and rapid penetration in the industry. Therefore, ensuring data security has become a real challenge for Big Data platforms. This work proposes a secure communication system through the implementation of a real scenario consisting of a set of software applications related to the acquisition, transformation and analysis of large amounts of information, also known as stack ELK. More precisely, in our proposal the data are sent encrypted from its source in workstations to their storage with encryption format in Elasticsearch, thus guaranteeing its confidentiality. The results show that the efficiency in the process of generation and delivery of data packets is not affected by the encryption process. The process to secure the messages information requires less than 2 millisecond per data packet, which meets the requirements of real-time monitoring.

CCS Concepts
• Security and privacy → Management and querying of encrypted data; Key management; • Computer systems organization → Availability.

Keywords
Big data; security information; vulnerabilities; threats.

1. INTRODUCTION
The data that is generated and collected in the different technological scenarios is on the verge of exceeding the imaginable limits of a digital universe increasingly threatened by the exponential increase in information produced at present [13].

As a consequence of this uncontrollable growth, big data enters the scene, as a set technology based tools on the analysis of structured, unstructured and semi-structured data generated in different sources of information [3]. Currently, the sources of information extend beyond the traditional structured databases, including services such as email, social networks, server logs, data generated by sensors, etc. Unstructured information that lacks a specific format [5]. The security risks to which organizations are exposed increase constantly, due to the fact that they face both internal and external threats, which is why it is necessary to safeguard the information considered as a critical asset in a company. Their damage or loss could interrupt activities, paralyzing services and causing serious economic damage [2].

Despite all the advantages of using Big Data due to the improvements in performance and productivity, it brings to a company, security problems represent a real challenge for this technology, much more when an organization processes and stores large amounts of information. This data can become the desired target by attackers and criminals. Nowadays, one of the most versatile tools to manage Big Data is Elasticsearch, which together with Logstash and Kibana make up the ELK stack, an open source platform that allows the analysis of data almost in real time on information collected from a source specific [8]. Elasticsearch was conceived with the main objective of analyzing large amounts of information in a distributed computing architecture. Since it is an open source tool with free access to the public, security was not a concern for its creators, which translates into security gaps in this tool on the information it stores.

This document examines the operation of the ELK stack, focusing especially on the transmission of information from the workstations to Elasticsearch, analyzing the level of security that this platform provides the stored data and providing an alternative solution to combat the deficiencies identified, contributing in this way in the safe and reliable use of this tool in the management of big data.

The rest of the paper is organized as follows. Section 2 presents the proposed architecture for the implementation of the ELK stack and its components. Section 3 analyzes the deployment and evaluation of the test platform, its particularities, configurations and review of data traffic, evidencing vulnerabilities in its transmission. In section 4 a solution model to the security gaps found is proposed. Section 5 analyzes the performance of the proposed solution by measuring the efficiency in the transmission of packets with data in clear and encrypted text, from their origin in the workstations to their destination in the Elasticsearch platform and in section 6 conclusions.

2. PLATFORM ARCHITECTURE
The Big Data platform is made up of four modules: data collection, transmission, storage and visualization as shown in Figure 1, which will allow us to perform the security analysis on the information circulating through the different stages of the platform.

ICBDE'19, March 30-April 1, 2019, London, United Kingdom

ACM ISBN 978-1-4503-6186-6/19/03…$15.00
DOI: https://doi.org/10.1145/3322134.3322154

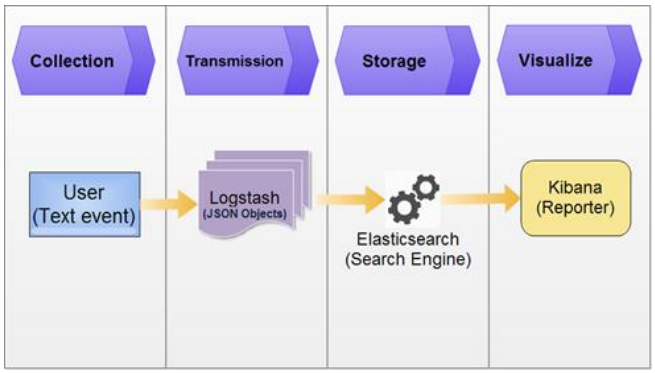

Figure 1. Platform architecture diagram.

2.1 Data Collection Module

Through the use of an agent installed in the work stations of the users, the extraction of data generated from the different sources of information that resides in these computers will be performed. This application is responsible for sending the data to the transport module who will be responsible for data collection.

2.2 Data Transmission Module

This module is responsible for the transmission of data generated by the agents, receives the information sent by the data collection module, processes the information, transforming it into a comprehensible format and then sending it to the data storage layer. For this purpose, we use Logstash as basic information processing software.

Logstash [9] is a tool that allows the transfer of information by collecting, processing and sending data to a specific destination. This powerful tool supports a wide variety of inputs and data processing such as filtering events, having as output a main destination like Elasticsearch.

This module is the meeting point of all the information sent by the data collection module, so it is necessary to consider faults in the transmission caused by a possible drop in service. For which an additional Logstash server is added to this module providing load balancing and high availability capabilities.

2.3 Data Storage Module

This module is responsible for storing the data generated by the agents located in the work stations. For this purpose, we used Elasticsearch [14], a Lucene based open source search engine designed to work in clusters by replicating data to other nodes providing uninterrupted service capability [10]. Also known as the heart of the ELK stack, it is a search platform almost in real time and works with the concept of inverted index allowing a fast insertion and recovery of data, in addition, it uses a method of multiple copies to guarantee the availability and reliability of stored data.

2.4 Display Module

This module performs the visualization of the data stored in Elasticsearch, providing a graphical interface for the analysis and search of information. Kibana [11] was used in the implementation of this module, a very versatile and intuitive opensource tool that interacts with Elasticsearch to perform the analysis and visualization of data.

3. DEPLOYMENT

3.1 Platform Deployment

The platform is implemented in a corporate network provided in its infrastructure of security equipment and a virtualized environment in which the solution has been deployed as can be seen in Figure 2. The workstations send the data generated by the different sources of information to the server Logstash who in turn will transmit this data to Elasticsearch for its storage and its later visualization in Kibana.

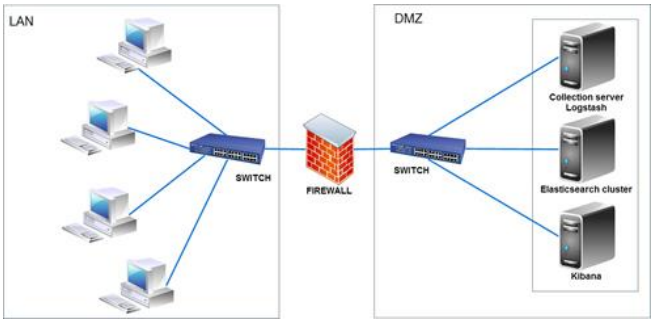

Figure 2. ELK deployment diagram.

3.2 Data Source And Collection

The information generated in the workstations is sent to the Logstash server through the use of an application created to detect any activity that a user performs on his computer, that is, this program will listen to everything the user types and sends it to the data collection module.

3.3 Data Transmission

This module collects the data sent by the agents installed in the work stations, providing load balancing and guaranteeing transmission reliability since it is configured in high availability. Logstash is made up of three components: input, output and filter, the latter formats the data according to certain specifications. The input component consumes the information coming from an information source and the output component sends the data to a specific destination [4]. The data is generally not structured and often contains inaccurate information that is not relevant for proper use, so the filter performs the analysis of the input fields and allows eliminating unnecessary information, as can be seen in Figure 3.

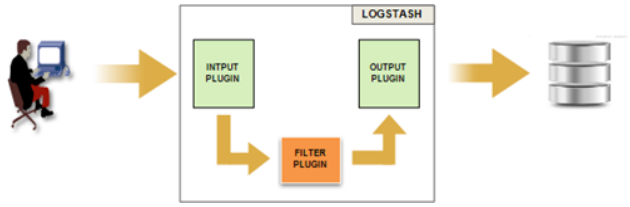

Figure 3. Data collection.

Logstash provides support for events and records generated by various network protocols, communication between processes, chat and email. It supports UDP, Websockets, HTTP and more. It has a plugin that allows messages to be read as events on the network through UDP and the only configuration field required for this plugin is the port, in our case we configure it with 5965 by which Logstash listens to events. It also uses a plain type codec that works with simple text without delimitation and the default

character encoding format is Unicode Transformation Format UTF-8, as can be seen in Figure 4.

```
input {
 udp {
  port => 5965
  codec => plain { charset => "UTF-8" }
  type => "TextEvent"
 }
}
```

Figure 4. Logstash input.

Once the information has been received and transformed into flat objects, Logstash using the grok filter analyzes the unstructured data and converts it into structured using regular expressions as can be seen in Figure 5, these patterns are incorporated in Logstash and allow to filter words, numbers and dates.

```
filter {
 if [type] == "TextEvent" {
  grok {
   break_on_match => false
   match => {
    message => "(?m)%{TIMESTAMP_ISO8601:sourceTimestamp} a:
%{GREEDYDATA:hostPrivateIP_b64} b:
%{GREEDYDATA:userDomain_b64} c: %{GREEDYDATA:agentId_b64} d:
%{GREEDYDATA:eventType_b64} - e:
%{GREEDYDATA:applicationTitle_b64} f:
%{GREEDYDATA:typedWord_b64}"
   }
  }
```

Figure 5. Logstash filter.

In the output we send Elasticsearch the information that will be stored in an index with the format "logstash-test-text-% + YYYY.MM.dd" as you can see in Figure 6. This will be the document that will contain the data collected by Logstash.

```
output {
 if [type] == "TextEvent" {
  elasticsearch {
   index => "logstash-test-text-%{+YYYY.MM.dd}"
   document_type => "TextEvent"
   hosts => "localhost"
  }
 }
```

Figure 6. Logstash output.

3.4 Data storage

The information coming from the distribution module, duly formatted and understandable, is received by Elasticsearch and stored in a "logstash-test-text-2019.01.13" index, as can be seen in Table 1. In addition, it can be identified that the information stored it is completely readable in clear text "Hello", which represents a serious security risk, so as a measure to guarantee the privacy of the information in the fields considered critical it is necessary to encrypt this data. Elasticsearch in its commercial mode has a tool that allows encrypting the stored information known as X-Pack [6], payment supplement that provides security to the data, but which in turn is restrictive for users who cannot access this resource for their high price. This work provides an alternative of free access to solve this problem, allowing to provide security by encrypting the data hosted in Elasticsearch.

Table 1. Information stored in clear text index

Field	Data
_index	logstash-test-text-2019.01.13
_type	TextEvent
_ sourceTimestamp	2019-01-12 17:55:02,082
_ typeWord	Hello

4. PROTECTING DATA

Big Data platforms should be able to host information in their nodes in a reliable and secure way, however, this data is exposed to threats that could damage its integrity. This research proposes a method that allows storing the information in an encrypted form using an encryption algorithm that allows us to provide security to the data generated in the collection module for later storage in Elasticsearch.

The techniques used to guarantee security in the Big Data platform are based on the Asymmetric Encryption Standard Advanced Encryption (AES) algorithm also known as Rijndael, which can process 128-bit blocks using encryption keys of 128, 192 and 256 bits [1], which allows to encrypt the information from its origin in the workstations until its storage in the cluster defined in Elasticsearch.

This method will provide protection against attacks that are intended to be used in the data collection and storage process in Elasticsearch with the objective of stealing sensitive and confidential information. This proposal is based on encrypting the communications that are generated from the endpoints by the agent providing an encryption algorithm to all the information generated by the end users.

The agent installed in the workstations was developed in C#, a programming language designed by Microsoft for its .NET platform [7], which is listening to any activity generated by a user collecting and transmitting through a channel insecure all the information entered, for this particular a program was designed using the Rijndel class located in the library System.Security.Cryptography, which allows to perform symmetric encryption or secret key encryption.

Once encrypted, the information is sent to Logstash, which in turn transmits it to Elasticsearch and as can be seen in Table 2, the stored information is encrypted, which guarantees the privacy of the information.

Table 2. Information stored in encrypted text index

Field	Data
_index	logstash-test-text-2019.01.13
_type	TextEvent
_ sourceTimestamp	2019-01-12 17:55:02,082
_ typeWord	37SrtQPZoVLQYgqCoJXexQ==

The data channel is listened to by capturing traffic using the Wireshark [12] traffic analysis tool, which allows to capture the live data on the network interface of the originating equipment, it is evidenced in accordance with Figure 7 that the information transmitted is encrypted, guaranteeing in this way that the data coming from the work stations is transmitted through the network in an secure manner, guaranteeing its confidentiality.

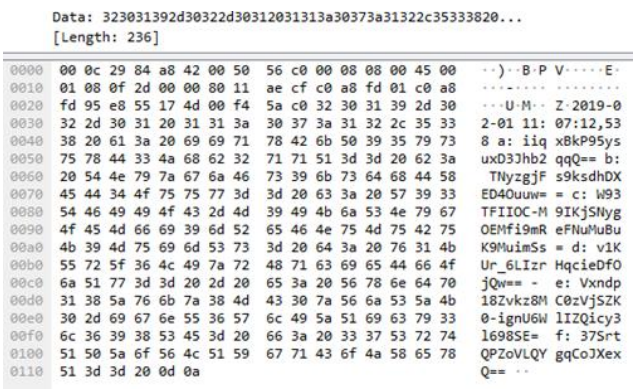

Figure 7. Wireshark encrypted text.

5. PERFORMANCE ANALYSIS

Once the proposed solution is implemented, it is necessary to validate that its operation does not affect the performance of the ELK platform, specifically the delivery time of information from the collection module to the storage module. Because we are doing the encryption of data this process could cause delays, so it is necessary to analyze the time it takes the information without encrypting from part of the work stations until your arrival at Elasticsearch, versus the time it takes encrypted information and identify if there are excessive delays.

The tests carried out consisted of measuring the response time by sending 4 groups of data packets in which each group contains 10 packages of the same size, the length of the packet is proportional to the content message which will increase upwards from one group to another. In the first group will be sent packages with the message "Hello", the second "Hello World", the third "Hello World Here" and the fourth "Hello World Here Now", from the workstations to the Elasticsearch server. This first scenario of experimentation was done without applying an encryption algorithm to then repeat the experiment with the encrypted messages.

Figure 8 presents the average response time of each data group. Noting that there is a difference of 2 milliseconds between group of packets with message in clear text and encrypted. Proving that the additional time required to encrypt the messages is imperceptible (2 milliseconds) does not compromise the performance of the platform, obtaining an adequate response and ensuring compliance with the requirements of monitoring in real time.

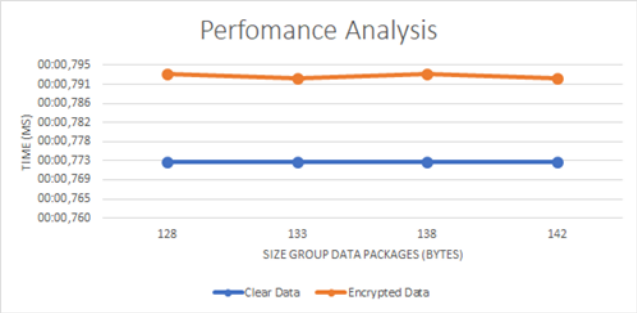

Figure 8. Performance analysis.

6. CONCLUSION

The handling of data from end users is sensitive, due to the importance that this information represents as personal keys, bank accounts, confidential information, etc. So, it is important to guarantee users privacy through the implementation of security mechanisms. Elasticsearch is a very powerful tool that provides many benefits for handling large amounts of information, but it only guarantees the confidentiality of information under the use of additional payment supplements to which not all users have access. So, in this paper a solution is designed and implemented to encrypt the information at source (work stations) which circulates through the network in a protected manner and is stored at the destination (Elasticserch) encrypted, thus guaranteeing the confidentiality of the information. The tests performed show that the process of encrypting the information does not compromise the performance of the ELK platform, demonstrating that the additional time required to encrypt the messages is imperceptible of less than 2 millisecond per data packet, obtaining an adequate response and ensuring compliance with the monitoring requirements in real time.

7. REFERENCES

[1] 2001. Advanced encryption standard (AES). Technical Report. DOI= https://doi.org/10.6028/nist.fips.197

[2] Ibrahim Yahya Mohammed Al-Mahbashi, M. B. Potdar, and Prashant Chauhan. 2017. Network security enhancement through effective log analysis using ELK. In *2017 International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. DOI= https://doi.org/10.1109/iccmc.2017.8282530

[3] Sahel Alouneh, Ismail Hababeh, Feras Al-Hawari, and Tamer Alajrami. 2016. Innovative methodology for elevating big data analysis and security. In *2016 2nd International Conference on Open Source Software Computing (OSSCOM)*. IEEE. DOI= https://doi.org/10.1109/osscom.2016. 7863685

[4] Dong Nguyen Doan and Gabriel Iuhasz. 2016. Tuning Logstash Garbage Collection for High Throughput in a Monitoring Platform. In *2016 18th International Symposium on Symbolic and Numeric Algo- rithms for Scientific Computing (SYNASC)*. IEEE. DOI= https://doi.org/10.1109/synasc.2016.063

[5] Chen He. 2015. Using Logstash and ElasticSearch to Achieve Real-time Statistical Analysis of DSpace Logs. *Data Analysis and Knowledge Discovery* 31, 5, Article 88 (2015), 5 pages.

DOI= https://doi.org/10.11925/ infotech.1003-3513.2015.05.12

[6] Elastic.co. 2018. Encrypting Communications. https://www.elastic. co/guide/en/x-pack/current/encrypting-communications.html# encrypting-communications.

[7] Begona Garcia, Amaia Mendez, Ibon Ruiz, and Javier Vicente. 2009. MultiPAS: JAVA, C++ and C# to Octave bridges. In *2009 IEEE Inter- national Conference on Acoustics, Speech and Signal Processing*. IEEE. DOI= https://doi.org/10.1109/icassp.2009.4960083

[8] Sasirekha GVK and Subramanyeswara Rao Dasari. 2016. Big Spectrum Data Analysis in DSA Enabled LTE-A Networks: A System Architec- ture. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE. DOI= https://doi.org/10.1109/iacc.2016.127

[9] Chen He. 2015. Using Logstash and ElasticSearch to Achieve Real-time Statistical Analysis of DSpace Logs. *Data Analysis and Knowledge Discovery* 31, 5, Article 88 (2015), 5 pages. DOI= https://doi.org/10.11925/ infotech.1003-3513.2015.05.12

[10] Jong-Hoon Lee, Young Soo Kim, Jong Hyun Kim, Ik Kyun Kim, and Ki-Jun Han. 2017. Building a big data platform for large-scale security data analysis. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. DOI= https://doi.org/ 10.1109/ictc.2017.8190830

[11] Tarun Prakash, Misha Kakkar, and Kritika Patel. 2016. Geo-identification of web users through logs using ELK stack. In *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*. IEEE. DOI= https://doi.org/10.1109/confluence.2016.7508191

[12] S Sandhya, Sohini Purkayastha, Emil Joshua, and Akash Deep. 2017. Assessment of website security by penetration testing using Wire- shark. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. DOI= https://doi.org/10.1109/ icaccs.2017.8014711

[13] Duygu Sinanc Terzi, Ramazan Terzi, and Seref Sagiroglu. 2015. A survey on security and privacy issues in big data. In *2015 10th Inter- national Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE. DOI= https://doi.org/10.1109/icitst.2015.7412089

[14] Urvi Thacker, Manjusha Pandey, and Siddharth S. Rautaray. 2016. Performance of elasticsearch in cloud environment with nGram and non-nGram indexing. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. IEEE. DOI= https://doi.org/10.1109/iceeot.2016.7755381