Journal of Information Security and Applications 75 (2023) 103509

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

Implementing and evaluating a GDPR-compliant open-source SIEM solution

Ana Paula Vazão [a], Leonel Santos [b], Rogério Luís de C. Costa [c,*], Carlos Rabadão [b]

[a] School of Technology and Management (ESTG), Polytechnic of Leiria, Leiria, 2411-901, Portugal
[b] Computer Science and Communication Research Center, School of Technology and Management (ESTG), Polytechnic of Leiria, Leiria, 2411-901, Portugal
[c] Computer Science and Communication Research Centre (CIIC), Polytechnic of Leiria, Leiria, 2411-901, Portugal

ARTICLE INFO

Keywords:
Security Information and Event Management
GDPR
Pseudonymization
Elastic stack

ABSTRACT

Security Information and Event Management (SIEM) solutions collect events from the IT infrastructure and concentrate information from the various components in a single place, allowing the detection of anomalous situations and attacks, and helping to protect confidential data. But real-world network environments may be complex and heterogeneous (e.g., in terms of devices, applications, and operating systems), and the attack surface can be vast, which makes increases the amount that a SIEM solution must collect and analyze. The General Data Protection Regulation (GDPR) has increased the level of complexity in such context, as organizations must ensure the monitoring of access to personal data and various levels of security in their infrastructure.

In this work, we deal with the implementation of an open-source SIEM solution that incorporates technical measures for the protection and control of personal data, ensuring compliance with the GDPR. We identify the main functionalities and describe a solution based on the Elastic Stack and additional open-source external tools.

To validate our proposals, we implemented a prototype of our solution in a real-world environment. We simulated internal and external attacks that show the solution capacity to deal in real-time with the detection of threats and incidents. We also evaluated the performance and resource consumption of personal data pseudonymization processes. Obtained results show our solution presents good performance and scalability.

1. Introduction

In the last decades, mobile and intelligent devices have enormously increased the volume of data generated and transmitted by computer networks. Maintaining security and privacy become even more challenging in the context of the smart networks [1]. Malicious actors are highly innovative, and cyberattacks have now become highly automated [2].

On the other hand, several regulations and legal requirements related to the processing and storage of personal data and that establish technical and organizational measures to protect privacy have entered into force in the last few years [3]. The European General Data Protection Regulation (GDPR) [4] is one of such regulations, which forces companies to reformulate their processes and implement various data (privacy) protection levels.

Security Information and Event Management (SIEM) solutions are extremely relevant in such context. They detect threats and security violations and contribute (together with tools like Intrusion Detection System/Intrusion Prevention systems, firewalls, and antivirus software) to data availability, integrity, traceability, and reliability.

But the implementation of a SIEM solution is quite complex. The attack surface can be vast, resulting in a large amount of data to be analyzed. The obligations imposed by data protection regulations make SIEM implementation even more complex, as they require the adoption of measures (e.g., pseudonymization) to protect personal data even at log level [5].

The main objective of this work is to define an open-source SIEM that incorporates technical measures for protecting and controlling personal data, ensuring compliance with the GDPR. Log data should be made available for querying without impacting the SIEM's incident detection performance. We identify the most appropriate techniques and procedures for personal data protection and define an architecture of a GDPR-compliant solution based on open-source software. We describe the pseudonymization algorithm and the main configurations to implement the GDPR-compliant open-source SIEM solution. To evaluate our proposals, we implemented a prototype and tested it in a real-world environment. We simulated internal and external attacks to evaluate the prototype's capacity to support real-time detection of threats and

* Corresponding author.
E-mail addresses: 2170101@my.ipleiria.pt (A.P. Vazão), leonel.santos@ipleiria.pt (L. Santos), rogerio.l.costa@ipleiria.pt (R.L.d.C. Costa), carlos.rabadao@ipleiria.pt (C. Rabadão).

https://doi.org/10.1016/j.jisa.2023.103509

Available online 17 May 2023

incidents. To assess the feasibility of the solution, we also measure performance and resource consumption during data pseudonymization.

In the next Section, we review some background and related work. Then, Section 3 describes the GDPR-compliant SIEM solution. In Section 4, we present experimental evaluation results. Finally, Section 5 contains the conclusions and proposals for future works.

2. Background and related work

Privacy and security of personal data to ensure compliance with the GDPR is also a key challenge for Organizations. It is necessary to ensure that personal data is protected and there is control of access to it. The GDPR has changed how organizations manage their IT infrastructures, as there is a requirement to implement additional procedures and security measures to ensure legal compliance of security systems.

SIEM is used to identify security incidents in real-time, contributing to compliance with the GDPR since it concentrates all the IT infrastructure logs and allows correlating, analyzing, and storing data from different sources. On the other hand, implementation and maintenance of a SIEM are costly, complex, and need constant updates to respond to organizations' evolution and the evolution and complexity of cyberattacks.

In a SIEM, it is essential to ensure the privacy of personal data collected, adopting protection mechanisms to defend them. As logs may contain personal data, the required security mechanisms may include log pseudonymization and access control.

2.1. Security information and event management

SIEM solutions integrate real-time security monitoring and information security management with the analysis of security logs and report generation [6]. One of the goals of a SIEM is to collect and concentrate information from the various security components in a single location, also ensuring centralized log management [6].

In essence, a SIEM solution provides a broad and detailed view of a company's security, enabling analysts to perform real-time security analysis [7]. The main technical requirements and uses of a SIEM solution include [8–14]:

- Log management, analysis and correlation;
- Forensic Analysis;
- File access auditing and compliance;
- Monitoring - including application log monitoring, user activity monitoring, file integrity monitoring, and device log monitoring;
- Threat intelligence feeds;
- Real-time alerting and incident response workflows;
- Dashboards and reporting.

In short, due to their characteristics, SIEM solutions enable earlier identification of cyberattacks and possible security incidents, thus contributing to the company's GDPR compliance [15].

2.2. General data protection regulation

In the last years, companies realized that personal data have intrinsic economic value. They started to make substantial investments in collecting and processing data from millions of European consumers. The General Data Protection Regulation regulates the use of personal data by creating a set of new rights, new procedures, and new obligations that all public or private entities must comply with, or they may suffer heavy monetary penalties. The GDPR came into force in May 2018, with entities given two years to implement it.

According to Article 4(1) of the GDRP, data that identifies its holder, or data that, although not directly identifying the person, allows easy identification of its holder are personal data [4,16]. It is a broad concept because it encompasses all information in any format that may allow the holder identification [16].

Several requirements and measures must be fulfilled when implementing a GDPR compliant SIEM [4,5,17]:

Table 1
Requirements fulfillment by open source and free SIEM solutions.

Requirement	Elastic Stack	Graylog	OSSIM	Splunk Free
Alerting	×	✓	✓	×
Anonymization	✓	✓	×	✓
Audit and monitor access to personal data	×	×	×	×
Authentication and security of user data	✓	✓	✓	✓
Compliance reporting	×	×	✓	×
Disaster recovery	✓	✓	×	×
Ensure protection by design and by default	×	×	×	×
Ensure resilience	✓	✓	×	×
File access auditing	×	×	×	×
File integrity monitoring	✓	×	✓	✓
Log analysis and correlation	✓	✓	✓	✓
Log management	✓	✓	×	✓
Monitoring	✓	✓	✓	✓
Notifications of data breaches	×	×	×	×
Reporting	×	×	✓	✓
Restrict access to personal data	✓	✓	✓	×
Scalability	✓	✓	×	×
Set retention time for data	✓	✓	×	✓
Threat intelligence feeds	×	×	✓	×
User activity monitoring	×	×	×	×

- Enable the pseudonymization of personal data;
- Allow retention times for personal data;
- Ensure the security of personal data;
- Make notifications of data breaches;
- Restrict access to personal data;
- Audit and monitor access to personal data;
- Ensure resilience;
- Ensure disaster recovery;
- Ensure protection by design and by default;
- Enable the creation of Compliance Reports.

2.3. Related work

There are some works on the use of open-source software to build SIEM solutions. In [18] authors compare the features of several SIEM solutions (including commercial versions). In [19], the author compares the use of Elastic Stack and Graylog to support a SIEM solution in an Internet service provider. The author concludes that the Elastic stack outperforms Graylog in terms of requirements fulfillment and usability.

Graylog and OSSIM are open source SIEM solutions. Elastic Stack is not a native SIEM, but has an open source edition that may be used together with other tools to build a SIEM system. Splunk is a SIEM that is not open source, but which has a *free* edition, named Splunk Free.

Table 1 [20] summarizes the fulfillment of the SIEM requirements and functionalities by the open source versions of Graylog, OSSIM and Elastic Stack, and by Splunk Free. The methodology used for building the table is described in [20]. None of those tools provides all the required functionalities, but Graylog and Elastic Stack are promising and scalable solutions that fulfill some requirements related to personal data protection [20]. Elastic Stack also allows the integration of other open-source or freeware tools, such as ReadonlyRest [21] or Search Guard [22] and also has a feature, named *fingerprint filter*, that may be used for data pseudonymization.

HELK [23] and ROCK [24] are projects that use components of Elastic Stack for log security. Varanda et al. [5] discuss the use of pseudonymization in the log processing pipeline for compliance with the GDPR. In [3], authors use Elasticsearch, Logstash, and Kibana log pseudonymization during log ingestion. Anonymize-it [25] is a general utility for anonymizing data. It may be run as a script that accepts a config file containing information regarding what to be anonymized.

Currently, the tool has two methods for anonymization, namely hash-based and faker-based anonymization. The first uses a unique user ID as a salt to anonymize fields, while the second uses the *Faker* Python package to create the fake data used to replace the original data. This method may lead to collisions and is not suitable for contexts in which one must maintain records linkability and global pseudonym consistency. In [25], the authors state that Anonymize-it 'is not intended to be used for anonymization requirements of GDPR policies'.

Menges et al. [26] present the conceptual architecture of a SIEM named DINGfest. Such architecture has four components: data acquisition, data stream, data analysis, and digital forensics & incident reporting. In [27], the authors extend the DINGfest architecture to deal with the protection of personal data. They aim to evaluate the extended solution both from a technical and legal perspective. Their approach is to delete personal data from collected data. In that work, the authors standardize the logs of the Linux operating system, enumerating four fields: *source*, *type_id*, *path*, and *misc*. They remove the path field (which may contain the user name in Linux systems) and use machine learning to identify incidents. The authors enumerate the possibility to apply the extended DINGfest architecture to an already established SIEM system, and the necessity of developing the data protection policy within SIEM systems.

Currently, there are several studies on using machine learning-based solutions that apply classification for attack identification, which include contexts like IoT (e.g., [28]), the analyses of encrypted traffic [29], Intrusion Detection Systems (IDS) [30], and SIEM [31]. Most of these works deal with the performance of the models. Indeed, machine learning performance is highly dependent on input data quality. Data normalization is one of the pre-processing techniques that most impact the performance of machine learning classification solutions [32]. But the full normalization of logs for machine learning consumption is challenging due to the wide variety of formats and the required normalization speed [33]. Indeed, the use of machine learning algorithms for log management in production environments faces the challenge of creating an anomaly detection solution that is fast, scalable, accurate and low-cost so that it can make it accessible to small and medium-sized businesses [34]. But to build a machine-learning-based GDPR-compliant SIEM, one must also deal with personal data protection. In this context, one possible approach would be to simply delete personal data from collected data (as in [27]). But that removes the possibility of using such data for legitimate purposes, like incident investigation and forensic analysis. Data anonymization may also be used to protect personal data, as it minimizes the risk of re-identification, but using data anonymization causes a deterioration in the performance of machine models, i.e., it reduces the models' efficiency to identify anomalies and attacks [35].

In this work, we use Elastic Stack components to build a GDPR-compliant SIEM and evaluate its log processing performance and behavior in environments with internal and external attacks. We present an algorithm to encrypt the sensitive data and replace it with the respective hash, without harming the detection of incidents. This approach takes advantage of the features already implemented in the SIEM system. The proposed solution also supports the re-identification of personal data for legitimate purposes.

3. GDPR-compliant open-source SIEM

In this section, we describe the proposed open-source SIEM solution, which guarantees the security of equipment and applications by allowing the identification of possible threats in real-time and at the same time pseudonymizing the sensitive data contained in the security logs.

3.1. Main functionalities

Our SIEM solution is based on the Elastic Stack and additional tools. It uses open and free software. Also, it contains technical measures to protect and control any personal information in log data, hence being a GDPR-compliant solution in protecting the privacy of sensitive data in logs.

The main functionalities of the proposed solution include functionalities *traditionally* related to SIEM systems:

- Collect data from different sources;
- Ensure compatibility with different operating systems;
- Standardize the logs of all components;
- Analyze and correlate the information of the equipment in real-time;
- Present several security dashboards and allow the customization of others;
- Create custom reports and alerts;
- Filter and highlight events by their criticality;
- Detect threats, security incidents, and vulnerabilities;
- Issue alerts in cases where there are suspicious activities on the network;
- Restrict access and enable multiple levels of permissions
- Provide multiple security mechanisms

But also include the following relevant technical measures for the protection and control of personal data:

- Enable the pseudonymization of personal data;
- Limit retention and allow the definition of retention times for personal data;
- Audit accesses to personal data;
- Restrict access to personal data;
- Ensure security of personal data;
- Ensure data integrity;
- Ensure data protection by design and by default.

3.2. Solution architecture

We use the Elastic Stack components (including Elasticsearch, Logstash, and Kibana) and open-source plugins (like ReadonyRest's Elasticsearch plugin) to build a SIEM solution.

Logstash collects the data from the beats and, through pipelines, transforms them and sends them to Elasticsearch. Elasticsearch is an analysis, search, and storage engine that receives the data from Beats and Logstash. Kibana is the solution's web interface for managing and viewing the data stored in Elasticsearch.

To collect the logs of the various constituent of a network, several types of Beats may be used, including Auditbeat, Filebeat, Heartbeat, Metricbeat, Packetbeat, and Winlogbeat. Some of them (e.g., Winlogbeat) are specific to certain operating systems.

The open source version of Elastic Stack does not provide the alerting functionality (as presented in Table 1). To add the functionality to our solution, we use the open-source framework Elastalert [36]. Elastalert creates alerts based on the information in Elasticsearch. To do this, this component periodically searches Elasticsearch and, if it finds a match with the defined rules, an alert is issued.

Pseudonymization will be ensured by Logstash during the data ingestion phase. We maintain re-identification data separately from pseudonymized data. Therefore, we use a second Elasticsearch server to store the index with personal data (original field value and the hash value). Hence, messages with pseudonymized personal data are sent from Logstash to a second Elasticsearch server where they can be searched and processed. The communication between Logstash and the second Elasticsearch server is encrypted.

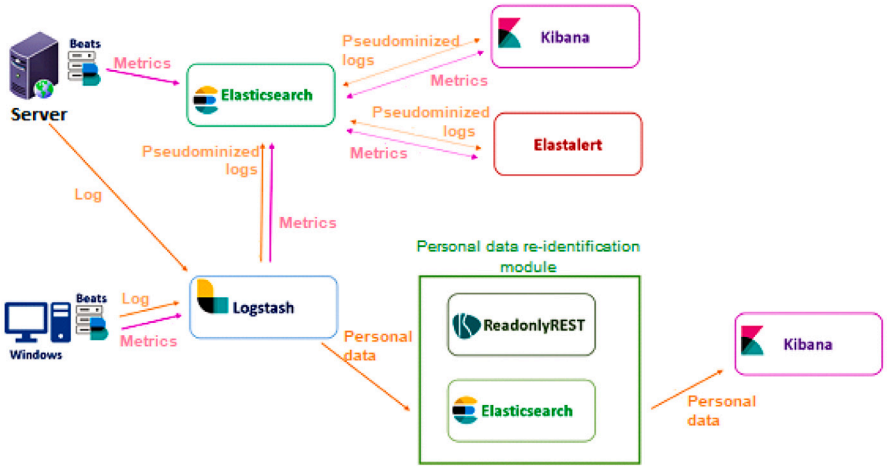


Fig. 1. Components of the open-source SIEM solution with pseudonymization.

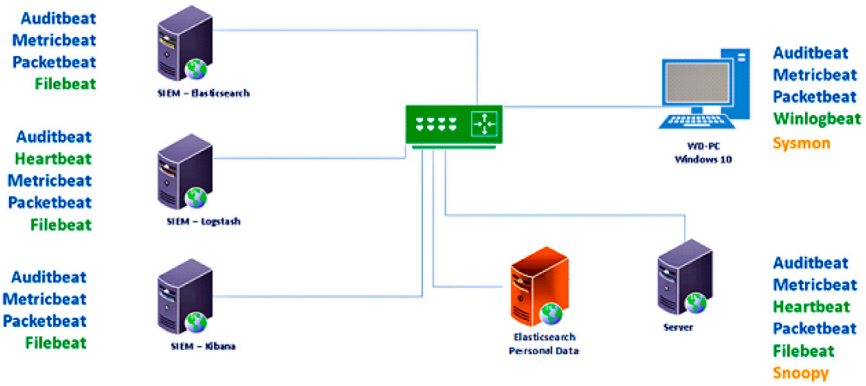


Fig. 2. Distribution of components between (physical andor virtual) machines with sample Beats.

This second server Elasticsearch is used together with a ReadonyRest plugin to create a personal data re-identification module, as represented in Fig. 1.

ReadonyRest audits the accesses to personal data stored in Elasticsearch, as it allows administrators to add a new security layer to the index where the fields that allow recovery are stored, i.e. through the information contained in the fields it is possible to identify the user and the machine responsible for a given occurrence.

In the proposed solution, re-identification is done manually and consists of searching the hashes that were identified in the attack on the Elasticsearch server with the pseudonymized data. Only authorized users may query re-identification data, and such queries are audited. Indeed, all actions performed in the index that stores the key that identifies the sensitive data are recorded in a separate index which permits presenting evidence in an audit process.

Fig. 2 presents an overview of the data flow between the solution components and exemplifies the use of beats on different operation systems. In Fig. 2 the machine running Windows 10 represents the workstations in the network, and a server machine represents all the servers in the network (e.g., web server and application server). Personal data in logs (e.g., equipment name, username, and IP addresses) is pseudonymized to make the solution compliant with the GDPR.

3.3. Log pseudonymization

We pseudonymize log data during the log ingestion phase. Algorithm 1 presents the overall log pseudonymization method. In such an Algorithm, the log file with data to be pseudonymized is the input.

After validating the type of log, we clone the log file. We use this clone to create the log with the pseudonymized data (we do not modify the original log). Consider that there is a set of generic fields (e.g., user name, hostname) whose data commonly is private and has to be pseudonymized (i.e., $Fields\_list$ in Algorithm 1). After cloning the log, we look for the existence of such fields in the cloned log. For each of those fields, we would look for the field value in the log and calculate its hash (e.g., using a salt, HMAC and SHA256) (line 9 in Algorithm 1).

Depending on the beat type, personal data maps to distinct fields. For instance, some fields that contain the hostname, user name, or IP address are:

- *message, agent.hostname, agent.name, host.hostname, host.name, host.ip* - in Auditbeat, Filebeat, Metricbeat, Packetbeat, and Winlogbeat;
- *user.audit.name, user.effective.group.name, user.effective.name, user.filesystem.group.name, user.filesystem.name* - in Auditbeat;
- *observer.hostname, observer.ip, monitor.ip* - in Heartbeat;
- *winlog.computer_name, winlog.event_data.SubjectUserName, winlog.event_data.TargetUserName, winlog.event_data.SubjectUserName* - in Winlogbeat.

The $Maps$ function in Algorithm 1 represents the mapping between generic personal data maps and beat specific fields. Then, we look for strings to be replaced by a hash in each identified mapping. In log files, fields like computer and user name commonly appear in different cases. Then, we compare the uppercase strings.

The original value of each pseudonymized field and the corresponding hash value are sent to an index hosted on a dedicated server. The

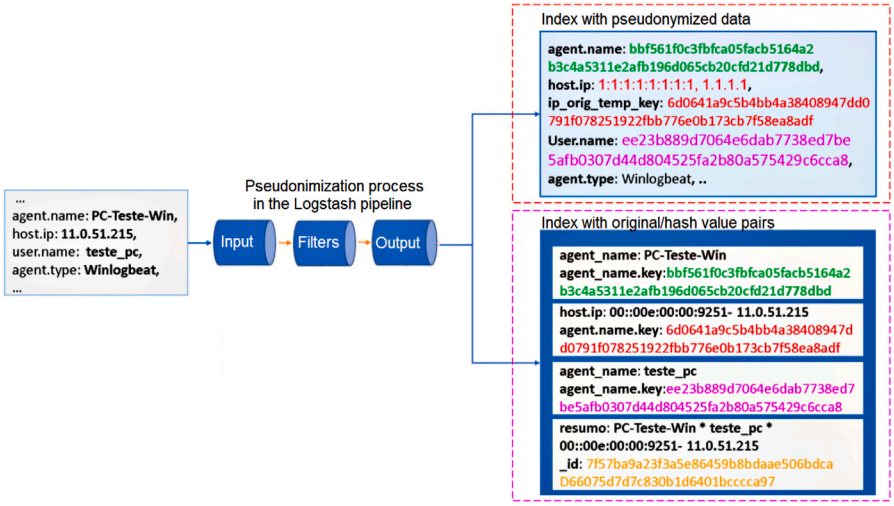


Fig. 3. Pseudonymization workflow with output sample.

Algorithm 1: Log pseudonymization

Input : *Log*: file with data to be pseudonymized;
Outputs: (i) Pseudonymized log sent to *Server_A*;
(ii) Original and hash value of each field sent to index in *Server_B*

Method
  Validate_log_type(*Log*);
  CLog ← *Clone*(*Log*);
  Fields_list ← list of generic fields that contain data to be pseudonymized;
  key ← 'HMAC-KEY' ;
  foreach *pfield* ∈ *Fields_list* **do**
    if *Exists*(*pfield*, *CLog*) **then**
      pfield_value ← *GetValue*(*CLog*, *pfield*) ;
      pfield_hash ← *Calculate_hash*(*pfield_value*, *key*, *HMAC*, *SHA256*);
      foreach *lfield* ∈ *CLog.fields* **do**
        if *Maps*(*lfield*,*pfield*) **then**
          if *Upper*(*pfield.value*) ⊆ *Upper*(*lfield.value*)) **then**
            Replace *lfield.value* in *Clog* with *pfield.hash*
          end
        end
      end
      SendValues(*pfield_value*, *pfield_hash*, *Server_A*)
    end
  end
  SendLog(*Clog*, *Server_B*)

log with the pseudonymized data is sent to the main server where users may query it.

Our solution uses the Logstash pipeline to pseudonymize log data. The Logstash pipeline has three main blocks: *input*, *filters* and *output*. The input block defines the event sources. Filters rules may be applied together with transformations to apply to filtered data. Flow control operators define the destination of each output, i.e., pseudonymized data is maintained apart from the index with the original and hash values. Fig. 3 presents an overview of the pseudonymization workflow. The implementation of Algorithm 1 is done through several configurations.

When the pipeline receives a log file, Logstash looks for fields to be pseudonymized. The *fingerprint filter* is used for identifying such filters and pseudonymizing them. Its configuration defines the field name and the hash method to be applied to the value of the field. This filter may use several hash methods (e.g., SHA1, SHA256, SHA384, SHA512, and MD5). The SHA256 offers a good balance between robustness and performance. Also, the use of salt is recommendable to improve security, then a key should be used. The system uses the HMAC function when one includes a key in the fingerprint filter configuration. The HMAC function is a robust method for creating hashes because it incorporates an additional secret using a key.

The following code presents an example of the configuration of the fingerprint filter. In such an example, the field *name* would be pseudonymized using SHA256. A key was also defined, which means that the HMAC function would be used.

```
fingerprint {
  source => "[agent][name]"
  target => "[@metadata][fingerprints]"
  method => "SHA256"
  key => "HMAC-KEY"
  id => fingerprint
}
```

The following code presents the use of the clone filter to make a copy of the log.

```
clone {
  clones => [ "clone_name_equip "]
  add_tag => [ "clone_name_equip"]
}
```

We use the prune filter with the whitelist option to keep certain fields and discard others from the cloned log. In the following code sample, all fields except the *host.ip* field, the *tags* field, and the *@timestamp* field are discarded. Then, the fields containing the original value/hash value are added. The original and hash values for the fields *name*, *user* and *ip_orig* are added. A summary field with all the values of the pseudonymized fields is created.

```
prune {
  interpolate => true
  whitelist_names => [ 'host.ip','tags', '
@timestamp' ]
  add_field => { "agent.name" => "%{[
@metadata][name]}" }
  add_field => { "agent.name_key" => "%{[
@metadata][fingerprints]}" }
```

```
    add_field => { "agent.user" => "%{[
@metadata][user]}" }
    add_field => { "agent.user_key" => "%{[
@metadata][fingerprintsus]}" }
      add_field => { "agent.ip" => "%{[
@metadata][ip_orig]}" }
      add_field => { "agent.ip_key" => "%{[
@metadata][fingerprintsip]}" }
      "summary" => "%{[@metadata][name]} * %{[
@metadata][user]} * %{[@metadata][ip_orig]}"
}
```

We use the mutate filter to deal with fields that appear in different cases (like computer and user name). For instance, the following code deals with user names in different cases. The mutate filter uppercases the field data and then make it possible to replace the original value with the hash value. Then, the hash value replaces the original value in the *message* field if there is a match (*gsub*).

```
mutate {
  uppercase => [ "[@metadata][user_uppercase
]" ]
  id => mutate_username_uppercase
}
mutate {
  gsub => [ "[message]", "%{[@metadata][
user_uppercase]}" , " %{[@metadata][
fingerprintsus]} " ]
  id => mutate_subs_msguser2
}
```

4. Experimental evaluation

We implemented a prototype of the proposed solution and evaluated it in a real-world environment. This section describes the validation scenarios and simulated attacks and presents the results obtained from the experimental evaluation, including performance metrics on log pseudonymization. Configuration files and scripts are available in https://github.com/AnaVazao/SIEM_GDPR.

4.1. Experimental environment

The prototype was implemented on the network of the company XLog.[1] The company XLog provides consulting and training in the area of technologies. The company is composed of three buildings (as represented in Fig. 4) connected by fiber optics and incorporates a datacenter, an antivirus, a firewall, and multiple servers that manage a diversity of services. It also has redundant Internet service. The network counts with almost 100 workstations, which use the Microsoft Windows operating system. Web and application servers use a Linux operation system. Its infrastructure provides two Wi-Fi networks, that are managed by a controller placed in the datacenter.

All machines used in the prototype scenario are virtual machines. There are five machines with the host operating system Microsoft Windows 10 and one with VMware ESXi 6.7. The servers run the Ubuntu operating system (various versions), and the client machines run the Microsoft Windows 10 operating system. Table 2 summarizes the main software and hardware used.

The main scenario of the SIEM prototype implements the technical measures for the protection and control of personal data along with the functionalities of a SIEM. However, to run performance tests and collect additional metrics, we also executed the SIEM prototype without the technical measures for personal data protection and control. The XLog's infrastructure was segregated into a test VLAN to create a test environment.

[1] The real name of the company is omitted due to privacy requirements

4.1.1. Security measures

Elastic Stack allows the security of its components to be implemented in stages: you can block unauthorized access to data, apply encryption to communications between nodes in a cluster and apply encryption outside the cluster. In our SIEM prototype, we implemented data encryption on all communications performed by the solution, ensuring that only authorized users access the solution's data.

ReadonyRest's Elasticsearch plugin is the open-source tool selected to implement technical measures to protect and control personal data. This plugin adds various levels of security to Elastic Stack and encrypts the data that is in transport between the different components (i.e., Beats, Logstash, Elasticsearch, and Kibana). Among the remaining features this plugin provides, we highlight the creation of groups and users, LDAP authentication, and information segregation [21].

Through ReadonlyRest's Elasticsearch plugin, it is possible to identify the user who performed a search request, as well as the date of access to the personal data. It was necessary to create users and define their permissions to audit the access to re-identification data. Thus, we created three users, one so that Logstash could communicate with Elasticsearch and two in Kibana. The user *elastic* has administrator permissions and can access all the Kibana options, while the user *userk* can read the Elasticsearch indexes and has writing permission on the audit index, since the intention was, at a later stage, to audit the operations performed on the data_key index. The user *userk* does not have permission to access the Kibana management menu.

It is necessary to activate the auditing functionality so that it is possible to audit the operations performed by users to the data_key index. In ReadonyRest's Elasticsearch audit configuration, one can define which indexes you want to audit, the name of the auditing file, and whether to save the search or not. The following are the settings made for auditing the data_key index.

```
enable: true
    audit_collector: true
    audit_include_query: ["data*"]
    audit_index_template: "'audit_logs'-yyy-
MM"
    audit_serializer: tech.beshu.ror.
requestcontext.QueryAuditLogSerializer
```

After configuring the settings in the *readonlyrest.yml* file that define the operations to be audited, the index is automatically created.

If the user *userk* needs to obtain the identification of the computer on which an incident occurred, he accesses the user_key index and searches for the hash value. Fig. 5 presents an example of an audit record containing the identification of the user who queried for the personal data, the time, and the source IP. Fig. 6 presents a search request performed by userk and the hash value the user-provided.

4.1.2. Security events collected

To evaluate the attack tracing, we should collect and analyze security events. In the Microsoft Windows 10 operating system, we used the events suggested by the LOG-MD tool [37], whose purpose is to help users decide which logs to activate and that provides a report listing if the events are activated. Sysmon v13.21 [38] was used and configured according [39], which is a optimized configuration to find suspicious behaviors [40]. The Elastic Stack solution already contemplates the use of the Sysmon tool and its logs can be collected by Beats.

From the Linux operating system, we used the Filebeat agent to collect the system logs. This agent is already an integral part of the Elastic Stack solution and enables the use of the Elastic Common Schema to normalize the logs of the most diverse components, thus enabling the correlation between the most diverse types of logs. We used the Snoopy Logger tool [41] to audit the command line on the Web Server.

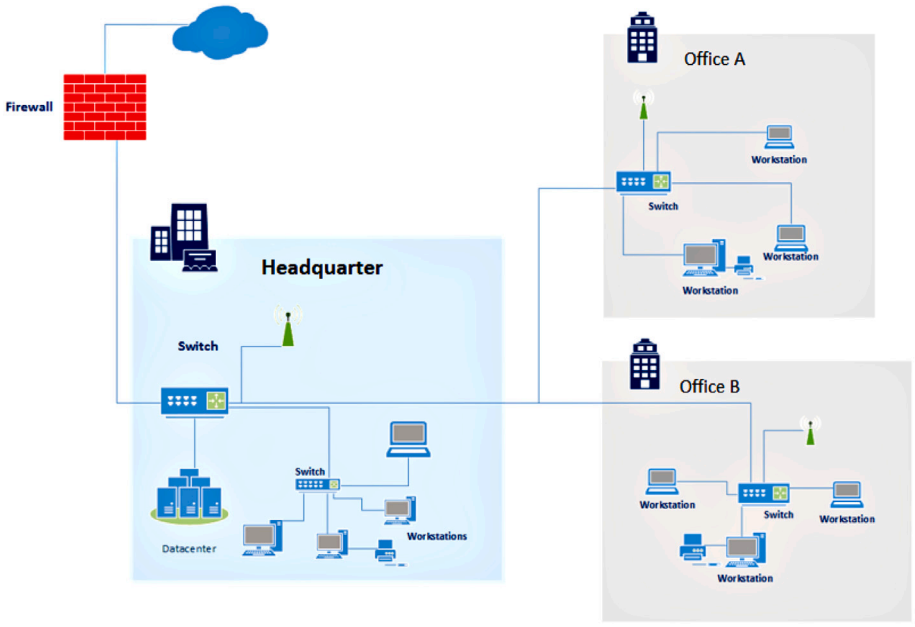


Fig. 4. XLog's network schema.

@timestamp	Oct 21, 2020 @ 13:40:35.000
_id	2011941103--148164270#13913
_index	audit_logs-2020-10
_score	1
_type	_doc
acl_history	❯ [Require HTTP Basic Auth-> RULES:[auth_key->false], RESOLVED:[indices=data*]], [L ->true], RESOLVED:[user=userk;indices=data*]]
action	indices:data/read/async_search/submit
block	{ name: 'Kibana User', policy: ALLOW, rules: [auth_key,actions]
content	❯ ⚠ {"version":true,"size":500,"sort":[{"@timestamp":{"order":"desc","unmapped_type": estamp","format":"date_time"}],"_source":{"excludes":[]},"query":{"bool":{"must": 0-21T23:59:59.999Z","format":"strict_date_optional_time"}}}],"should":[],"must_no
content_len	820
content_len_kb	0
destination	32
final_state	ALLOWED
headers	Connection, Content-Length, content-type, Authorization, Host
id	2011941103--148164270#13913
indices	
match	true
origin	/32
path	/data*/_async_search
processingMillis	0
req_method	POST
task_id	13,913
type	SubmitAsyncSearchRequest
user	userk

Fig. 5. Audit result of an operation performed by *userk*.

Table 2
Experimental setup description.

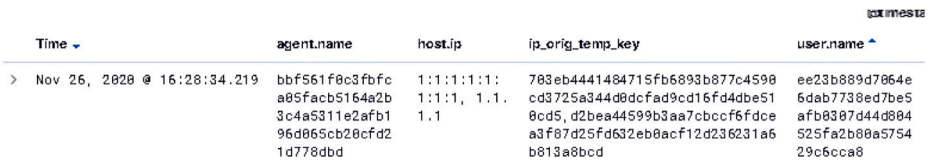
Equipment	Virtualization software/host operation system	Virtual machine	Software
Desktop Intel(R) Core(TM) i3-8100 CPU @ 3.60 GHz, RAM 16 GB, SSD 240 GB, HDD 1 TB, HDD 4 TB	VMware ESXi™ 6.7	8 GB RAM 500 GB HDD Ubuntu 18.04 LTS	Elasticsearch, Elastalert, Netdata, Auditbeat, Filebeat, Metricbeat, Packetbeat
		5 GB RAM 200 GB HDD Ubuntu 18.04 LTS	Logstash, Netdata, Auditbeat, Filebeat. Heartbeat, Metricbeat, Packetbeat
Desktop Intel(R) Core (TM) i7-3770 CPU @ 3.40 GHz, RAM 12 GB, SSD 240 GB, HDD 500 GB	Oracle VM VirtualBox Manager 6.1/ Microsoft Windows 10 Pro	8 GB RAM 200 GB HDD Ubuntu 18.04 LTS	Kibana, Netdata, Auditbeat, Filebeat, Metricbeat, Packetbeat
Desktop Intel(R) Core (TM) i5CPU 760@ 2.80 GHz, RAM 6 GB, HDD 1 TB	Oracle VM VirtualBox Manager 5.2/ Microsoft Windows 10 Pro	3 GB RAM, 50 GB HDD	Auditbeat, Metricbeat, Packetbeat, Winlogbeat, Sysmon
Desktop Intel(R) Core (TM) i5CPU760@ 2.80 GHz, RAM 6 GB HDD 2 TB	Oracle VM VirtualBox Manager 5.2/ Microsoft Windows 10 Pro	4 GB RAM, 120 GB HDD, Ubuntu 18.04 LTS	Web server, Netdata, Auditbeat, Filebeat, Heartbeat, Metricbeat, Packetbeat, Snoopy
Portátil Asus – BU403UA Intel(R) Core (TM) i7-6500U CPU@ 2.50 GHz, RAM 12 GB, SSD 240 GB, HDD 1 TB	Oracle VM VirtualBox Manager 5.1/ Microsoft Windows 10 Pro	3 GB RAM, 130 GB HDD Kali 2020.3	Pupy
Desktop Intel(R) Core (TM) i7-3770 CPU @ 3.40 GHz, RAM 16,0 GB, SSD 500 GB	Oracle VM VirtualBox Manager 6.1/ Microsoft Windows 10 Pro	6 GB, 200 GB SSD, Ubuntu 20.04 LTS	Prometheus, Grafana
		7 GB, 200 GB, Ubuntu 20.04 LTS	Elasticsearch, Kibana, ReadonlyRest

Fig. 6. Detail of the data stored in the audit index when accessing the index containing personal data.

4.1.3. Pseudonymization configuration and compliance with the GDPR

We configured the prototype to pseudonymize the hostname, source IP address, and user fields. We used SHA256 and HMAC in the pseudonymization. But the data type of the IP field is not compatible with such configuration. Then, we changed the IP address field to a fixed value (i.e., it was filled with the number "1") and a new field with the IP address hash value was added to the log. The new field was named ip_orig_temp_key. Through the ip_orig_temp_key field, it is possible to relate the IP address that is in the index with the original field and its hash value. The *mutate* filter performs several operations in such process, including adding a field (*add_field*), coping a value (*copy*), and removing a field (*remove_field*).

The following is an excerpt of the code that performs these operations.



Time	agent.name	host.ip	ip_orig_temp_key	user.name
Nov 26, 2020 @ 16:28:34.219	bbf561f0c3fbfc a05facb5164a2b 3c4a5311e2afb1 96d065cb20cfd2 1d778dbd	1:1:1:1:1: 1:1:1, 1.1. 1.1	703eb4441484715fb6893b877c4590 cd3725a344d0dcfad9cd16fd4dbe51 0cd5, d2bea44599b3aa7cbccf6fdce a3f87d25fd632eb0acf12d236231a6 b813a8bcd	ee23b889d7064e 6dab7738ed7be5 afb0307d44d884 525fa2b80a5754 29c6cca8

Fig. 7. Pseudonymized field: machine name, User, and IP address.

```
if [host][ip] {
  mutate {
    copy => { "[host][ip]" => "ip_orig_temp"
 }
  }
  mutate {
    split => [ "ip_orig_temp" , "," ]
    add_field => { "ip_testes" => "%{[
ip_orig_temp][1]}" }
  }
  mutate {
    replace => { "[@metadata][ip_orig]" =>
"%{[ip_testes]}" }
  }
  fingerprint {
    source => "[host][ip]"
    target => "[@metadata][fingerprintsip]"
    method => "SHA256"
    key => "HMAC-SHA-256"
  }
  #replace the IP address value with the
hash value if not copying
  if "clone_name_equip" not in [tags] {
    mutate {
      add_field => { "new_host_ip" => [
"1:1:1:1:1:1:1:1:1" , "1.1.1.1" ] }
    }

    # create the field where the hash value
will be stored
    mutate {
      add_field => { "ip_orig_temp_key" =>
"%{[@metadata][fingerprintsip]}"}
    }
    mutate {
      copy => { "new_host_ip" => "[host][ip
]" }
    }
    mutate {
      gsub => [ "%{"[@metadata][ip_orig]}" ,
 " %{[@metadata][fingerprintsip]} " ]
    }
    mutate {
      remove_field => [ "new_host_ip" , "
ip_orig_temp" , "ip_testes" ]
    }
  }
}
```

The result of the pseudonymization can be seen in Fig. 7. A record was created in the data_key index for each log processed by the pipeline. As a result, the index had thousands of identical records. To reduce the index size, we used the fingerprint and the *MURMUR3* method (a hash function that uses multiplication and rotation operations and guarantees that there are no duplicate values in the index) to create a unique id.

It was also possible to assign various levels of permissions to the users created, either through the Elastic Stack solution or through ReadonlyRest's Elasticsearch plugin. One can also define which users may query a particular index and what operations they may perform.

Besides the various security mechanisms previously described, we also used HTTPS (HyperText Transfer Protocol Secure) on our prototype. Another security feature that Kibana provides is the possibility to set the retention times for the data stored in the indexes. Data in Elasticsearch can assume four phases, as identified below: Hot, Warm, Cold, and Delete, and it is possible to specify the retention time for data at each phase.

When a log is firstly sent to Elasticsearch, its sets the @version field to "1". For any write operation performed on the log, Elasticsearch increments the version number by one. This version control is a key feature because it ensures log integrity.

4.2. Simulation of attacks

In an internal attack, attackers fraudulently obtain valid credentials and use them in actions that seem legitimate and authorized. In external attacks, attackers do not have the required credentials, so their activities are usually more intrusive.

In our experiments, we simulated an internal attack and two external attacks. For the internal attack, we used the Pupy tool (from the MITRE ATT&CK Framework [42]), while for the external attacks we used the Hydra tool (made available by Kali [43]) and the Simple-SYN-Flood script.[2]

4.2.1. Internal attacks

Pupy is a cross-platform (Microsoft Windows, Linux, and Android) remote access tool that allows the management of several computers simultaneously, the establishment of SSH connections, and the integration with Mimikatz (a well-known tool for credential extraction on Windows operating systems).

Initially, we created the client payload and compressed it using a password. Then, the compressed file was uploaded to a Gmail Drive. An attacker would have to convince a user who has administrator rights to run the uploaded file, ignoring any antivirus alert. When the file is executed, the "attacker" gains access to the user's computer.

Using the SIEM prototype, it is possible to visualize the communications between the attacker's machine and the attacked one (using the Timelines feature of the Elastic Stack).

The password hashes may be used to discover weak passwords. We downloaded the hashes of all users through the *creddump* command. Then, using the SIEM, we tried to find pieces of evidence of the *creddump* command. Such traces were found when analyzing the logs through the Kibana's Discover option. Fig. 8 shows the evidence of this action. We point out that the user responsible for this operation is the user who executed the payload *client.exe*.

We also simulated an attack using Pupy and Mimikatz. The Microsoft Defender antivirus blocks the operation, and this time it was not allowed to be executed, so the attacker was unable to obtain the credentials. However, it was possible to identify evidence through the SIEM of the attempt to obtain credentials. It is noteworthy that the log does not contain the tool's name, but there are indications that an attempt to escalate privileges to obtain credentials occurred.

Pupy has many functionalities and from the range of possibilities offered, the one that allows access to the command line of the victim's

[2] Available on GitHub https://github.com/Leeon123/Simple-SYN-Flood.

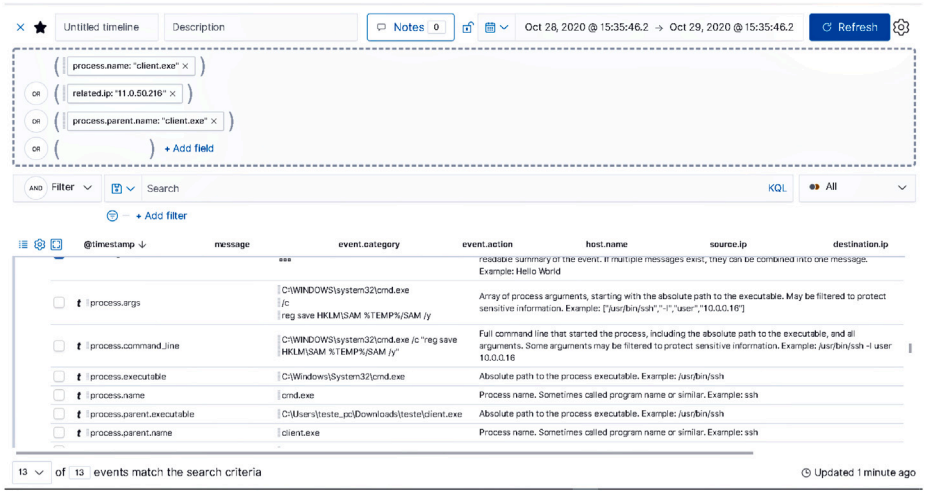


Fig. 8. Example of the logs created with Pupy's creddump command.

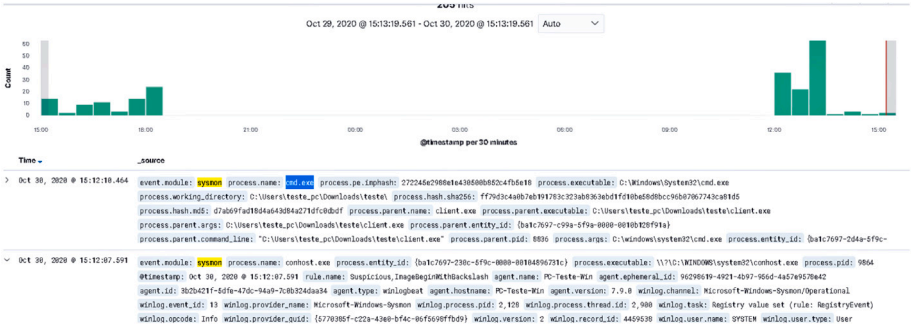


Fig. 9. Execution of cmd.exe by Pupy that created a new command line.

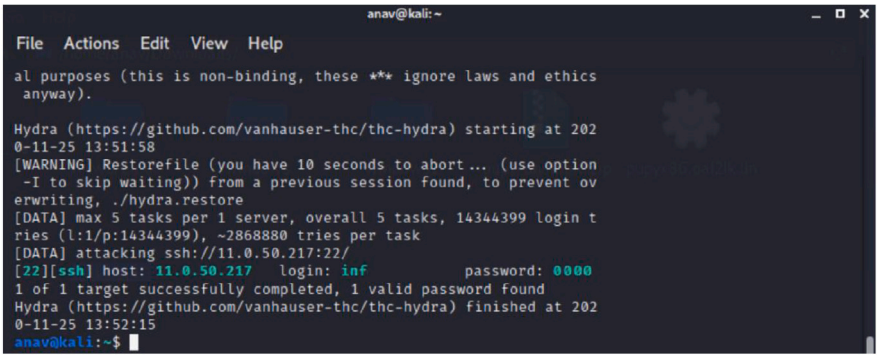


Fig. 10. Getting credentials through Hydra.

computer was selected. It was possible with the Shell command to view all the folders and files and execute commands.

Fig. 9 presents the logs created by Sysmon. It was possible to find in the logs that the command *cmd.exe* was used to create a new command line and that its parent process is the payload *client.exe*.

4.2.2. External attacks

We simulate two types of external attacks: brute force and denial of service.

Using the Hydra tool, we simulated a brute force attack on the SSH service of the webserver. We configured the webserver with a weak password and, then, it was possible to obtain the access credentials to the victim machine, as can be seen from the data obtained and available in Fig. 10.

In the Elastalert, we created a rule to raise alerts for SSH login errors. Through the SIEM, it was possible to verify the occurrence of 2466 authentication attempts to the SSH service, as shown in Fig. 11.

Using the Simple-SYN-Flood script, we simulated an SYN Flood denial of service attack. This attack aims to exhaust or block the services and, then, prevent access by other users. To simulate the denial of service attack, we accessed the Web server remotely through SSH, downloaded the script file, and executed it. Fig. 12 shows the SIEM dashboard, where the data flow transmitted to the Web server during the attack is represented.

Through the tests carried out, it was possible to validate the use of the SIEM prototype in tracking the "footprints" of attacks. Sysmon's log collection functionality was useful in classifying the operations performed on the server as resulting from attacks.

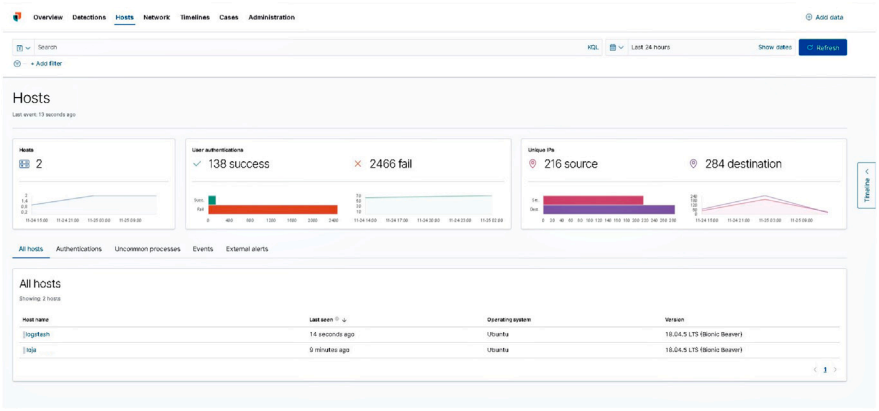


Fig. 11. SIEM: Authentication attempts to the SSH Service.

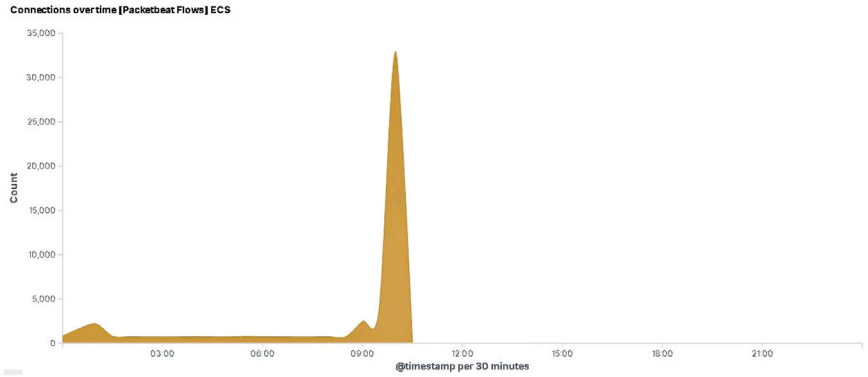


Fig. 12. SIEM: Dashboard with flow evolution over time.

Table 3
Metrics of the Logstash pipeline with and without pseudonymization.

	Events received	JVM Heap (MB)	Output events	CPU (%)
1 log	–	199.9/MB	–	2%
100 logs	–	184.6/MB	–	2%
1000 logs	75/s	203.7/Mb	100.1/s	91%
10,000 logs	387.5/s	201.0/MB	387.5/s	94%
1 log pseudonymized	–	201.5/MB	–	60%
100 logs pseudonymized	–	223.9/MB	–	95%
1000 logs pseudonymized	25/s	208.0/MB	33.37/s	93%
10,000 logs pseudonymized	220.83/s	286.4/MB	220.83/s	91%

4.3. Pseudonymization performance

We tested two scenarios to measure the cost of pseudonymization, one in which the logs are pseudonymized and another in which they are not. To make a fair comparison, we used the same snapshot of virtual machines in both scenarios, i.e., they had identical initial configurations.

In each scenario, we used four Comma-separated values (CSV) files. The first one contains a manually created row. To build the remaining three files, we copied the selected log data 100, 1000, and 10 000 times, respectively. The files were sent to the SIEM at distinct times (we chose a time in a way that log processing does not overlap with the processing of other data).

We collected several load and performance metrics in both scenarios. Table 3 presents the results in terms of the number of events received per second, the Java Virtual Machine (JVM) Heap size, the number of events emitted per second by Logstash, and CPU utilization.

Through the values measured during the experiments, it was possible to verify that an increase in the volume of data to be processed does not lead to an increase of the same proportion in resources consumption. This is mainly due to the way the JVM Heap distributes the log processing over time, i.e., the period required to process the logs that were pseudonymized was longer than those that were not. The log ingestion took longer than the pseudonymization process.

Also, comparing the scenarios with and without pseudonymization, it is possible to verify that the number of events received/issued per second is higher when there is no pseudonymization. Also, it is possible to see that the RAM consumption did not have a very significant increase when pseudonymizing log data.

4.4. Discussion

Hence, we experimentally verified through running a prototype in a real-world company that our proposal fulfills the requisites of a GDPR-compliant SIEM solution. We evaluated the data collected from different sources and operation systems, normalized logs from distinct components, analyzed and correlated equipment information in real-time, customized dashboards and alerts, validated the detection of threats and incidents, and ensured the security of personal data, executing data pseudonymization, restricting accesses according to multiple levels of permissions and auditing all accesses to personal data, between others.

The fulfillment of the functionalities listed in Section 3.1 may be verified by:

- Collect data from different sources — In our prototype, Beats were used to collect logs from Microsoft Windows and Linux operating systems. Beats is also compatible with various applications, such as Zabbix, MySQL, and Oracle.

- Ensure compatibility with different operating systems — Elasticsearch, Kibana, and Logstash were installed on the Linux operating system, but these components are compatible with Microsoft Windows and macOS. Beats were installed on Linux and Microsoft Windows operating systems.
- Standardize the logs of all components — Using the Elastic Common Schema functionality, it is possible to normalize logs from different sources, allowing them to be correlated. We used the ECS to standardize the log files collected from operating systems.
- Analyze and correlate equipment information in real-time — Kibana provides the Timelines feature in its Security option. We used such a feature to track the logs created by the attack.
- Present several security dashboards and allow the customization of others — The solution already provides a large number of security dashboards, and it was also possible for users to create their dashboards.
- Create custom reports and alerts — Kibana provides multiple dashboards which may create such reports. One may also export the search results to CSV. We incorporated the Elastalert tool in the SEIM prototype architecture to create alerts.
- Filter and highlight events by their criticality — One may search the Sysmon events to trace the attack logs. Fig. 9 presents an example of such search results, with the searched term highlighted.
- Detect threats, security incidents, and vulnerabilities — Through the tools provided by Kibana, it was possible to trace the actions performed by the Pupy tool, Hydra, and the Simple-SYN-Flood script.
- Issue alerts in cases where there are suspicious activities on the network — Elastalert allowed the creation of multiple rules. For instance, in our experiments, we created a rule to raise an alert whenever there were three failed authentication attempts in one hour.
- Restrict access and enable multiple levels of permissions - Through Kibana, it was possible to define several levels of permissions. Through ReadonlyRest's Elasticsearch plugin, it was possible to restrict access and assign various levels of permissions.
- Provide multiple security mechanisms — There are several security mechanisms, such as authentication, several permissions levels, encryption, and auditing index accesses.
- Enable the pseudonymization of personal data — Pseudonymization is performed through the Logstash pipeline. Fig. 7 exemplifies the pseudonymization results.
- Limit retention and allow the definition of retention times for personal data — One can set retention times for data using the index lifecycle management feature in Elasticsearch.
- Audit accesses to personal data — With ReadonlyRest's Elasticsearch plugin, it was possible to audit the operations performed by authenticated users on the index containing the recovery data, as exemplified in Fig. 5.
- Restrict access to personal data — In our experiments, we created three users to which we assigned different permission levels using ReadonlyRest's Elasticsearch plugin.
- Ensure security of personal data — In the SIEM prototype, we performed several operations to limit, audit, and ensure personal data security. Some examples are the pseudonymization, encryption, and auditing of the operations performed on the index that holds the recovery keys.
- Ensure data integrity — Data integrity is preserved through encryption and various levels of permissions. Elasticsearch manages the log version, and on any write operation performed on the log, Elasticsearch increments the version number by one (i.e., @version="1" means that the log file has not been modified).
- Ensure data protection by design and by default — The SIEM prototype ensures data security throughout its lifecycle. However, in commercial editions of Elasticsearch, there are more robust security options, and it is also possible to receive alerts about anomalies or set permissions at the field level.

We also assessed resource usage by the pseudonymization process. The results obtained demonstrated that the open-source prototype is scalable, thus efficiently handling an increasing amount of data.

5. Conclusions and future work

The entry into force of the General Data Protection Regulation has reinforced the relevance of Information Security measures. Entities that process personal data must prove that they implement the appropriate technical measures for data protection and control. SIEM solutions are extremely relevant in such context, as they support threats and security violations detection and contribute to data integrity, traceability, and reliability. In this work, we deal with the implementation of a SIEM solution based on open-source software that is GDPR-compliant in terms of log pseudonymization.

We identified the main functionalities of a SIEM solution and the appropriate technical measures for data protection and control. We describe the use of the Elastic Stack together with additional open-source plugins in a SIEM solution. We implemented a prototype of the proposed solution and evaluated its use in a real-world environment.

One of the measures referenced in the regulation is the pseudonymization of personal data. Hence, we evaluated the use of the prototype to pseudonymize personal data identified in security logs and the prototype's ability to audit accesses to the indexes that have re-identification data. We simulated internal and external attacks and also measured resource consumption due to the pseudonymization process.

Results prove the solution's ability to detect threats and incidents, and that it ensures the security and privacy of personal data. Also, we showed that the proposal handles an increasing number of data without an impact in the same proportion on resources' consumption.

As future work, we plan to study the incorporation of machine learning-based threat identification into the solution.

CRediT authorship contribution statement

Ana Paula Vazão: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Leonel Santos:** Conceptualization, Methodology, Writing – original draft, Supervision, Project administration. **Rogério Luís de C. Costa:** Investigation, Writing – original draft, Writing – review & editing, Visualization. **Carlos Rabadão:** Conceptualization, Methodology, Writing – original draft, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is partially funded by national funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., Portugal, under the Scientific Employment Stimulus - Institutional Call - CEECINST/00051/2018 and in the context of the project UIDB/04524/2020.

References

[1] Hamad SA, Sheng QZ, Zhang WE, Nepal S. Realizing an internet of secure things: A survey on issues and enabling technologies. IEEE Commun Surv Tutor 2020;22(2):1372–91. http://dx.doi.org/10.1109/COMST.2020.2976075.

[2] European Union Agency for Cybersecurity (ENISA). Research topics - ENISA threat landscape. Technical Report, 2020, p. 20.

[3] Varanda A, Santos L, Costa RLdC, Oliveira A, Rabadão C. Log pseudonymization: Privacy maintenance in practice. J Inf Secur Appl 2021;63:103021. http://dx.doi.org/10.1016/j.jisa.2021.103021.

[4] The European Parliament and the Council of the European Union. Regulation (EU) 2016/679 of the European parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. Official J Eur Union 2016;(59):1–88.

[5] Varanda A, Santos L, Costa RLdC, Oliveira A, Rabadão C. The general data protection regulation and log pseudonymization. In: International conference on advanced information networking and applications. Springer; 2021, p. 479–90. http://dx.doi.org/10.1007/978-3-030-75078-7_48.

[6] Ciampa M. CompTIA security+ guide to network security fundamentals. Cengage Learning; 2021.

[7] Watts S. IEM vs log management: What's the difference?, BMC. 2018, URL https://www.bmc.com/blogs/siem-vs-log-management-whats-the-difference/, Last accessed November 10, 2021.

[8] Catescu G. Detecting insider threats using security information and event management (SIEM) (Ph.D. thesis), UAS Technikum Wien; 2018.

[9] El Arass M, Souissi N. Smart SIEM: From big data logs and events to smart data alerts. Int J Innov Technol Explor Eng 2019;8(8):3186–91.

[10] Fortinet. What Really Matters When Selecting a Security Information and Event Management Solution. 2021, URL https://www.fortinet.com/content/dam/fortinet/assets/ebook/eb-security-information-event-management.pdf, Last accessed November 10, 2021.

[11] Mokalled H, Catelli R, Casola V, Debertol D, Meda E, Zunino R. The applicability of a siem solution: Requirements and evaluation. In: 2019 IEEE 28th International conference on enabling technologies: Infrastructure for collaborative enterprises. WETICE, IEEE; 2019, p. 132–7.

[12] Stefanova D. SIEM solutions and data protection compliance. 2020, URL https://logsentinel.com/blog/siem-solutions-and-data-protection-compliance/, Last accessed November 10, 2021.

[13] The Graylog Blog. SIEM, simplified. 2018, URL https://www.graylog.org/post/siem-simplified/, Last accessed November 10, 2021.

[14] Vacca JR. Computer and information security handbook. Newnes; 2012.

[15] Vazão A, Santos L, Piedade MB, Rabadão C. SIEM open source solutions: A comparative study. In: 2019 14th Iberian conference on information systems and technologies. CISTI, IEEE; 2019, p. 1–5.

[16] Team IGP. EU general data protection regulation (GDPR)–An implementation and compliance guide. IT Governance Ltd; 2020.

[17] Elasticsearch. GDPR compliance & the elastic stack. 2018, p. 13, URL https://www.elastic.co/pdf/white-paper-of-gdpr-compliance-with-elastic-and-the-elastic-stack.pdf, White paper WP-GDPR050218. Last accessed April 26, 2022.

[18] DiSIEM – Diversity-enhancements for SIEMs. In-depth analysis of SIEMs extensibility. Technical Report, DiSIEM consortium; 2017, p. 148.

[19] Bělousov P. Security enhancement deploying SIEM in a small ISP environment. Master's thesis, Faculty of Business and Management - Brno University of Technology; 2019.

[20] Vazão A, Santos L, Oliveira A, Rabadao C. A GDPR compliant SIEM solution. In: European conference on cyber warfare and security. Academic Conferences International Limited; 2021, p. 440–XIV.

[21] ReadonlyREST. ReadonlyREST for Elasticsearch. 2021, URL https://docs.readonlyrest.com/elasticsearch, Last accessed November 11, 2021.

[22] Guard S. Security and alerting for Elasticsearch by Search Guard. 2021, URL https://search-guard.com/, Last accessed November 11, 2021.

[23] Rodriguez R. The hunting ELK repository. 2021, URL https://github.com/Cyb3rWard0g/HELK/, Last accessed November 11, 2021.

[24] RockNSM. ROCK NSM - response operation collection kit. 2019, URL http://rocknsm.io/, Last accessed November 11, 2021.

[25] Elastic. anonymize-it. 2022, URL https://github.com/elastic/anonymize-it/, Last accessed August 04, 2022.

[26] Menges F, Böhm F, Vielberth M, Puchta A, Taubmann B, Rakotondravony N, Latzo T. Introducing dingfest: An architecture for next generation siem systems. In: Langweg H, Meier M, Witt BC, Reinhardt D, editors. Sicherheit 2018. Bonn: Gesellschaft für Informatik e.V.; 2018, p. 257–60. http://dx.doi.org/10.18420/sicherheit2018_21.

[27] Menges F, Latzo T, Vielberth M, Sobola S, Pöhls HC, Taubmann B, Köstler J, Puchta A, Freiling F, Reiser HP, et al. Towards GDPR-compliant data processing in modern SIEM systems. Comput Secur 2021;103:102165.

[28] Prazeres N, Costa RLdC, Santos L, Rabadão C. Evaluation of AI-based malware detection in IoT network traffic. In: Proceedings of the 19th International conference on security and cryptography, Vol. 1. Science and Technology Publications; 2022, p. 580–5. http://dx.doi.org/10.5220/0011279600003283.

[29] Di Mauro M, Di Sarno C. Improving SIEM capabilities through an enhanced probe for encrypted skype traffic detection. J Inf Secur Appl 2018;38:85–95. http://dx.doi.org/10.1016/j.jisa.2017.12.001, URL https://www.sciencedirect.com/science/article/pii/S2214212617303307.

[30] Prazeres N, de C. Costa RL, Santos L, Rabadão C. Engineering the application of machine learning in an IDS based on IoT traffic flow. Intell Syst Appl 2023;17:200189. http://dx.doi.org/10.1016/j.iswa.2023.200189, URL https://www.sciencedirect.com/science/article/pii/S2667305323000145.

[31] Moukafih N, Orhanou G, El Hajji S. Neural network-based voting system with high capacity and low computation for intrusion detection in SIEM/IDS systems. Secur Commun Netw 2020;2020:1–15.

[32] Singh D, Singh B. Investigating the impact of data normalization on classification performance. Appl Soft Comput 2020;97:105524. http://dx.doi.org/10.1016/j.asoc.2019.105524, URL https://www.sciencedirect.com/science/article/pii/S1568494619302947.

[33] Jaeger D, Cheng F, Meinel C. Accelerating event processing for security analytics on a distributed in-memory platform. In: 2018 IEEE 16th Intl conf on dependable, autonomic and secure computing, 16th intl conf on pervasive intelligence and computing, 4th intl conf on big data intelligence and computing and cyber science and technology congress(DASC/PiCom/DataCom/CyberSciTech). 2018, p. 634–43. http://dx.doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00114.

[34] Collier R, Azarmi B. Machine learning with the elastic stack: expert techniques to integrate machine learning with distributed search and analytics. Packt Publishing Ltd; 2019.

[35] Senavirathne N, Torra V. On the role of data anonymization in machine learning privacy. In: 2020 IEEE 19th International conference on trust, security and privacy in computing and communications (TrustCom). 2020, p. 664–75. http://dx.doi.org/10.1109/TrustCom50675.2020.00093.

[36] Yelp. Easy & flexible alerting with ElasticSearch. 2020, URL https://github.com/Yelp/elastalert/, Last accessed April 26, 2022.

[37] IMF Security LLC. LOG-MD discover it. 2021, URL https://www.imfsecurity.com/free, Last accessed November 24, 2021.

[38] Russinovich M, Garnier T. Sysmon - windows sysinternals. 2021, URL https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon, Last accessed November 24, 2021.

[39] SwiftOnSecurity. A Sysmon configuration file for everybody to fork. 2021, URL https://github.com/SwiftOnSecurity/sysmon-config, Last accessed November 24, 2021.

[40] O'Leary M, O'Leary M, McDermott. Cyber operations. Springer; 2019.

[41] Jese B. Snoopy Logger. 2021, URL https://github.com/a2o/snoopy, Last accessed November 24, 2021.

[42] MITRE ATT&CK. Pupy Software. 2021, URL https://attack.mitre.org/software/S0192/, Last accessed November 24, 2021.

[43] The Kali Team. Hydra Kali Linux Tools Software. 2021, URL https://www.kali.org/tools/hydra/, Last accessed November 24, 2021.