TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

ITC70LT
Maarja-Liisa Tammepõld 162975IVCM

SECURING THE CENTRALIZED LOGGING SYSTEM BY THE EXAMPLE OF ELASTICSEARCH

Master's thesis

Supervisor: Toomas Lepik
MSc

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

ITC70LT
Maarja-Liisa Tammepõld 162975IVCM

KESKSE LOGIMISSÜSTEEMI TURVALISEMAKS MUUTMINE ELASTICSEARCHI NÄITEL

Magistritöö

Juhendaja: Toomas Lepik
MSc

Tallinn 2020

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis and this thesis has not been presented for examination or submitted for defence anywhere else. All used materials, references to the literature and work of others have been cited.

Author: Maarja-Liisa Tammepõld

19.05.2020

Abstract

This thesis solves the security issues that comes with the default installation of the Elasticsearch server. The author finds ways to secure Elasticsearch data from Kibana and directly from Elasticsearch. Its focus has been to find an open source security solution that has a wide variety of authentication methods with a role-based access control using encrypted channels with configuration examples that would suit for different companies.

Firstly, the author gives an overview of information that needs protection and importance of securing personal data. Then the author takes a brief overview of NoSQL security implementations in MongoDB, OrientDB, Apache CouchDB and Elasticsearch between different licenses. After that the author gives an overview of companies' Elasticsearch use cases and examples how they have been using it and why companies prefer using Elasticsearch. Then the author describes Elastic Stack and gives an overview of logging solution alternatives and states why to continue with Elasticsearch in her thesis. Also, the author gives an overview of biggest data breaches with Elasticsearch that companies have recently faced. Based on previous information, the author raises requirements for security tools and starts comparison between them.

The author compared Open Distro Security, Search Guard, ReadonlyRest, Apache authentication and Elasticsearch basic license with X-Pack features in the table and then tested the most suitable solutions in her testing environment to choose the best solution according to the stated testing questions.

The author chose Open Distro Security for Elasticsearch to be the most suitable solution that suits for many companies, includes different authentication methods and user roles and is free to use for everyone.

This thesis is written in English and is 99 pages long, including 9 chapters, 3 figures and 5 tables.

Annotatsioon

Keskse logimissüsteemi turvalisemaks muutmine Elasticsearchi näitel

Lõputöös parandatakse andmete turvalisust puudutavaid probleeme, mis tuleb kaasa Elasticsearch'i paigaldamisel vaikimisi konfiguratsiooniga. Autor otsib lahendusi, kuidas kaitsta Elasticsearch'is olevaid andmeid läbi Kibana ning Elasticsearchi päringute. Keskendutakse vabavaralise lahenduse leidmisele, mis kasutaks erinevaid autentimismeetodeid läbi krüpteeritud kanalite ning rollipõhiseid õiguseid. Tuuakse välja konfiguratsiooninäidiseid, mida saaksid ettevõtted enda lahendustes kasutada.

Esimeses osas annab autor ülevaate infost, mis vajab kaitset ning räägib selle olulisusest. Autor teeb lühiülevaate erinevate NoSQL andmebaaside turvameetmetest, kus käsitletakse MongoDB, OrientDB, Apache CouchDB ja Elasticsearch'i erinevatest litsentsidest tulenevaid turvameetmeid. Tuuakse välja, mille poolest ettevõtted on valinud endale Elasticsearch'i ning kirjeldatakse Elasticsearch'i kasutusvõimalused. Tuuakse näiteid, kes kuidas Elasticsearch'i kasutab ning kirjeldatakse lahti Elastic Stack. Vaadatakse otsa logilahenduse alternatiividele, mis turul on ning mille poolest Elasticsearch'iga töös jätkatakse. Antakse ülevaade ka suurimatest viimase aja andmeleketest, mis Elasticsearch'i andmebaasi kasutamisega on esinenud. Peatükk lõppeb Elasticsearch'i turvarakendustele püstitatud nõuetega eesoleva info põhjal ning alustatakse nendevahelist võrdlust.

Autor võrdles tabelis omavahel Open Distro Security'it, Search Guard'i, ReadOnlyREST'i, Apache autentimist ning Elasticsearch'i Basic litsentsi X-Pack erisustega. Seejärel valis välja kõige paremini sobivad lahendused ning testis neid oma testkeskkonnas testküsimuste alusel.

Autor valis välja Open Distro Security Elasticsearch'i turvalisemaks muutmiseks, mis võiks sobida paljudele ettevõtetele, kus saab endale sobiva autentimismeetodi valida mitmete valikute hulgast, luua erineva tasemega kasutajarolle ning on vabalt kõigile tasuta saadaolev lahendus.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 99 leheküljel, 9 peatükki, 3 joonist, 5 tabelit.

List of abbreviations and terms

ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certification Authority
CI/CD	Continuous Integration and Continuous Delivery
CLI	Command Line Interface
CVE	Common Vulnerabilities and Exposures
DES	Data Encryption Standard
DN	Distinguished Name
EDPS	European Data Protection Supervisor
EOL	End of Life
GDPR	General Data Protection Regulation
GUI	Graphical Unit Interface
HTTP	Hypertext Transfer Protocol
ICO	UK's Information Commissioner's Office
IT	Information Technology
KDC	Kerberos Key Distribution Center
LDAP	Lightweight Directory Access Protocol
LTS	Long Term Support
OEM	Original Equipment Manufacturer
OS	Operating System
PII	Personally Identifiable Information
RBAC	Role-based Access Control
SCRAM	Salter Challenge Response Authentication Mechanism
UI	User Interface
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

Table of contents

List of figures

List of tables

1 Introduction

Logs give valuable information about a system state. They present a system health, different application queries and authentications. Developers use them for debugging applications, system administrators use them for understanding system health and troubleshooting problems, security specialists use them for analysing security-related events and monitoring anomalies, the business side can use data for better decisions. To optimize data sharing many organizations deploy centralized log collection systems that come with various shapes and sizes.

The diversity of users and information creates the need for compartmentalization, but the nature of logs makes it difficult. It is possible if the log collecting environment exist, so that it would provide access to logs based on the need and minimum principle. The goal is to find an open source solution for a centralized logging system, where it is possible to share access to specific log information according to the log fields using strict group policy and the need for information.

It is important to secure access to logs, so that information would not be lost, and relevant people could see information from logs. The aim is to protect collected data and user privacy, so that those would not leak. If more systems would implement proper security mechanisms, users would trust IT systems more.

By rumours, this kind of commercial solutions that support such assortment of logs cost a lot of money that smaller and medium businesses could not afford. Furthermore, companies that use CI/CD [Continuous Integration and Continuous Delivery] pipelining, where the development cycle is fast and the need for separate logging systems is growing to avoid compatibility issues, could not afford to buy licences for every logging system to secure the logs. What is more, some companies have had incidents where centralized logging systems were accessible from public networks without any security implementation. These are outlined in this thesis in paragraph 4.2.6.

To choose the solution that many companies could use, the author looked at the opensource solutions.

To analyse the open source software, the author uses qualitative comparative analysis. Based on the analysis' results with the case study method, the author chooses best ones out of them, so that the analysis would be thorough enough. The applied observational study is used to test how the solutions work in practice intending to find a solution for securing access to logs. The author takes into account the best security guidelines that have been suggested in the literature and will compare different open source software to secure log collection. According to the requirements, the author installs them in her test environment and compares them practically as stated in her testing questions to choose the best solution. Methodology in addition is described more specifically in the chapter 3.

The main contributions of this thesis are:

- Analysis and tests for Elasticsearch applications.
- Configuration examples of Security tools.
- Recommendations for securing Elastic Stack.

In this thesis the topics are aligned as follows:

- Chapter 2 gives an overview of related work with the thesis.
- Chapter 3 includes a description of the methodology of the thesis in more detail.
- Background information of logging and analysis starting point can be found in chapter 4. It gives an overview of the importance of logging and about the information that needs protection. Moreover, it gives an overview of security of NoSQL databases and differences between Elasticsearch licenses, Elasticsearch use cases and why companies have chosen Elasticsearch for them. Then, the author investigates Elastic Stack more specifically, searches if there are good alternatives for Elastic Stack and justifies why Elastic Stack is chosen to be improved. After that, the author gives an overview of biggest data breaches of Elasticsearch.

- Analysis for securing Elasticsearch continues in chapter 4: stating requirements for securing Elasticsearch, describing security tools for Elasticsearch, comparing them in the table and choosing solutions for testing.

- Chapter 5 gives an overview of the testing environment and explains how the author tested different security solutions for Elasticsearch data: Open Distro Security, Search Guard, Elasticsearch 7.6 with X-Pack.

- Chapter 6 gives an overview of the tests and final recommendations for securing Elasticsearch.

- Chapter 7 gives an idea for additional data protection in the centralized logging server that may be needed for future work.

2 Related work

Concrete thesis is a specific engineering task that concentrates on user access to Elasticsearch NoSQL database to protect logs from unauthorized access.

This thesis [1] has written about providing reliable log delivery and integrity of logs. Its author concentrated on how to provide reliability of logs using syslog daemons like rsyslog, syslog-ng Premium Edition and Fluentd and used Guardtime KSI for signing and verifying logs to proof log integrity. In conclusion he stated that Guardtime KSI is the best solution for providing log integrity and it can be integrated to rsyslog [1]. What is more, from the research came out that the default configuration can influence data loss on log delivery and specific configuration can make difference on log transportation. The least data loss on log delivery had Fluentd with its default configuration [1]. But using specific configuration, no data loss was found on any of the solutions that was tested [1]. This can be used in the authors' centralized log system. For example, if all rsyslog logs are collected to one collector server without any data loss, then it can be sent to Elasticsearch server using Logstash and later in Elasticsearch analyse logs without worrying if some data might be lost.

Another thesis [2] brought out the most common mistakes done in logging and gave instructions on how to remove them. As a result, its' author had a detailed requirement document for logging [2]. This thesis gave guidance what kind of data should be included in the logs. If logging is set properly then clients can send data to hosts with all the valuable information that is needed for services. As this contains a lot of different data and specific values that are useful for a different group of users depending on their work, log data distribution is needed and can be solved with the proper tool for authentication and authorization in Elasticsearch server. For example, to separate security logs, debug log, error log, user query information from one log file by different fields.

This paper [3] focuses on NoSQL databases, specifically analyses Elasticsearch and Apache CouchDB. It aims to transfer the dataset from Relational database to NoSQL

databases and analyses CouchDB and Elasticsearch performance. It states that CouchDB is more efficient than Elasticsearch during insert, update and delete operations but in the selection operation Elasticsearch performs better [3]. Logs require a lot of disk space because one message in a log file can be long and logs are written often, which can make relation database slow. Thus, it is important to do queries fast enough to have the possibility to analyse information in big datasets. It is a good knowledge that selection operation is better in Elasticsearch than in Apache CouchDB.

Another example of fast speed of searching in Elasticsearch is from thesis [4] which investigated performance optimization methods that can be applied to the database systems and compared MySQL Percona, MongoDB Percona, Splunk and Elasticsearch with an objective to provide rapid search results on leaked data with limited hardware specifications. Research has stated that Elasticsearch is the fastest for searching leaked data with an average response time of 1.58 seconds on data sets containing between 10 and 100 million records [4].

A paper about a security pattern for key-value NoSQL database authorization [5] represented a model that protects each individual cell in NoSQL key-value databases. It labels them with authorization rights following an RBAC [Role-Based Access Control] model or similar. The research paper presented a pattern to describe an RBAC model that exists in several Big Data systems. They concluded that MongoDB uses an RBAC model to protect collections of documents, which are heterogeneous, hierarchical records composed of sets of key-value pairs [5].

This paper [6] documented the definition of the standard security features to investigate the area of standard database security. It had a selection of NoSQL databases that consisted of OrientDB, Redis, Cassandra and MongoDB, which were used in a description of systematic investigation of standard database security features. This article found that NoSQL databases have not implemented proper security by default, and security implementation should be improved in them. It found that default configurations can be done with no passwords or default users with default passwords. Also, no role and permission management exist, ports are without protection and accepting connections from all clients. This article pointed out that in NoSQL databases default configuration is not secure and needs additional configuration.

This paper [7] develops a mechanism to handle large scale data processing and searching operations using sharding technique. The results of a paper interpret that sharding technique provides the efficient mechanism to handle large scale of data and states that sharding technique is an efficient mechanism to process big data which provides better scalability and fault-tolerance. Its results indicate that as the number of shards increased the similarity score of query and document also increases, the most relevant document is retrieved for a given query. This kind of technique makes searching quicker. Sharding increases search performance through smaller index sizes. "As data is distributed in multiple shards each shard will have smaller index size which result in faster processing and enhances the document search performances [7]."

Document about the performance of ELK stack and commercial system in security log analysis [8] found that ELK stack shows similar or better performance in searching for particular security logs that matches the specific condition. "For 1,000 million log files, ELK stacks took 1 min and 14.4 sec, whereas 1 min and 22.2 sec with Splunk" [8].

A paper [9] about a solution for the secure use of Kibana and Elasticsearch in multi-user environments concentrates on protecting own data from others or sharing part of data among a group. Its authors have found that the CERN cloud service group had provided a cloud utilization dashboard to each user by Elasticsearch and Kibana with restrict data access based on a user authenticated by the CERN Single Sign on system. They have proposed an alternative to that system, which would not be useful only for cloud services and have found out that a good alternative for user and group-based access control for Elasticsearch is a Search Guard and have described the effect on performance with using Search Guard. Their result shows that indexing throughput performance is degraded 20% with Search Guard.

This paper [10] is about building a real-world logging infrastructure with Logstash, Elasticsearch and Kibana. It concentrated on how to set up Elasticsearch, Logstash and Kibana on a dedicated virtual machine. Logstash was configured to consume logging messages from log4j. Also, it includes automating the process of backing up Elasticsearch contents and removing outdated records. Moreover, configures an alerting mechanism that sends notifications to email. In this paper, the logging infrastructure was successfully built from scratch using ELK, but its author found also that Elastic business model does not provide a complete set of tools for free.

3 Methodology

This thesis has a technical problem that needs a practical solution. This chapter explains the methodological approach of the thesis.

Qualitative analysis methods have been chosen to answer research problem. Qualitative methods are used for describing, interpreting, contextualizing, and gaining in-depth insight into specific concepts or phenomena [11].

Information is needed that could give in-depth insight into a topic, that leads to the best solution and solves the research problem. The aim is to have a practical solution that can be used in different companies which use Elasticsearch and Kibana in their centralized logging system.

Information was gathered from internet publications, articles, blogs and software owners' websites from Google and Google Scholar. The most recent and relevant materials were chosen. Also, the author questioned product vendors to get information that was not presented on the web.

Research is based on the information written in English. Content analysis will be used to categorize and discuss the meaning of words, phrases and sentences [11]. Case study will be used to get concrete, contextual, in-depth knowledge about a specific real-world subject and to describe, compare, evaluate and understand different aspects of a research problem [11].

Case study allow going into detail about a specific event [12]. It enables to explore the concepts and reasoning around why something is the way it is [12].

Moreover, the applied observational study will be used to test how the solutions work in practice. It can be used if the access to an operational environment exists and is used to execute a comparative study that demonstrates the results and to determine how effective and different the solution performs in comparison to other solutions [13].

Furthermore, the snowball method will also be used to gather more information from literature by using a key document on a subject as a starting point [14].

In this thesis, tables are used to compare solutions and rate their features.

The analysis consists of the following steps:

1. Describing why logging is important and giving an overview of information that needs protection
2. Finding out how different NoSQL databases have secured themselves and searching the alternatives to Elastic stack.
3. Describing Elastic Stack and its use cases.
4. Finding out the biggest data breaches with Elasticsearch.
5. Describing requirements for choosing the best solutions.
6. Describing comparable products.
7. Comparing solutions in the table.
8. Communicating with vendors if all the information cannot be found in the web.
9. Choosing the best solutions for testing.
10. Describing the testing environment and raising questions for testing.
11. Testing solutions to investigate them further and find the best solution.

4 Background information and analysis

This chapter gives a brief overview of the information that needs protection, discusses different NoSQL database security mechanisms and goes more specific to Elastic Stack components and its security implementation. It describes the centralized logging server of Elastic Stack and gives an overview of the biggest Elasticsearch data breaches. The chapter ends with the requirements for Elasticsearch security tools and analyses different security solutions for Elasticsearch and its' components. Open Distro Security, Search Guard, ReadOnlyREST, Elasticsearch basic license with X-Pack features and Apache authentication are compared.

4.1 Importance of logs

Information systems grow like the requirements for logging. Specialists are not able to view hosts by host all the system logs. They need a central logging server that includes all the information of their systems. Logging is not only diagnosing and troubleshooting issues, but also for monitoring systems. There are tools that are worked out to get critical business metrics and data from logs [15].

Log file can include information about system errors, debug information, user session data, IP addresses, user queries of HTTP body, authentication attempts, successful and failed activities [2]. This data can be collected from different services and sent to a centralized logging server for analysation and for storage. The amount of different information about services arises interest in various user groups that want to access that data depending on their job responsibilities. Even one log file can include information that needs to be cut out before sharing its' data. This kind of information that needs to be removed before sharing can be user session data and IP addresses. The bigger the systems are, the more administrative tasks come of these systems and the need for using tools that help to administrate the systems securely, so that information can be shared more easily.

The next paragraph investigates information that should not be shared out if user do not have the need for that information.

4.1.1 Information that needs protection

Logging is important and different teams have various purposes to view the logs. To ensure that right people will get to the information that they need, a proper group policy must be implemented.

Firstly, what kind of information needs to be protected from unauthorized access to safeguard the privacy or security of an individual or organisation [16]?

The main types of sensitive information are:

1) Sensitive personally identifiable information, data that can be traced back to an individual and could harm that person [16]. GDPR [General Data Protection Regulation] states that the sensitive personal data that need higher level of protection is information that includes genetic, biometric and health data, also racial and ethnic origin, political opinions, religious or ideological convictions or trade union membership [17].

2) Business information, data that can pose a risk to the company if discovered by a competitor or general public. This information includes trade secrets, acquisition plans, financial data, supplier and customer information [16].

3) Classified information, data that belongs to a government and is restricted according to level of sensitivity [16].

Besides that, general personal data, also known as PII [Personally Identifiable Information], that need to be protected according to GDPR, data that can identify an individual includes for example a person's name, identification number, location data, online identifier or special characteristics, which expresses the physical, physiological, genetic, mental, commercial, cultural or social identity of a person. Also, all data which are or can be assigned to a person in any kind of way, like the telephone, credit card or personnel number of a person, account data, number plate, appearance, customer number or address. Furthermore, all the data that can identify a person, like personal

working time, breaks, IP addresses. Also, subjective information like opinions and judgements can be personal data [17].

GDPR is a regulation that was formulated to protect personal data and privacy in EU [European Union] countries. It regulates the processing by an individual, a company or an organisation of personal data relating to individuals in the EU and applies for uses of personal data outside the personal sphere, for example, for sociocultural or financial activities. It does not apply to the processing of personal data of dead persons or of legal persons [18].

EDPS [European Data Protection Supervisor] that is an independent EU body, is responsible for monitoring the application of data protection rules within European Institutions and for investigating complaints. Data protection officer independently ensures the internal application of data protection rules are in cooperation with EDPS in European institutions [19].

GDPR violation brings fines to offenders. For example, French data protection regulator imposed a 50 million EUR fine to Google for alleged infringements of the transparency principle and lack of valid consent [20]. Another example is from July 2019, when ICO [UK's Information Commissioner's Office] issued a 204,6 million EUR fine to the British Airways for a violation of Article 31 of the GDPR, when the British Airways website diverted users' traffic to a hacker website, which resulted in stealing the personal data of more than 500 000 customers, because of the poor security mechanisms [21].

Security mechanisms are important and if these are not implemented correctly and it is related to personal data loss, GDPR fines might be assigned.

In the next paragraph differences of NoSQL and SQL databases have been outlined and then an overview of NoSQL database security implementations will be given.

4.2 NoSQL databases

Based on research done in [22] author formed the Table 1.

The main differences between NoSQL and SQL databases are given in the Table 1 below. The structure is not the only difference between NoSQL and SQL databases.

Table 1. Differences between NoSQL and SQL databases.

NoSQL	**SQL**
Non-relational or distributed database.	Relational database.
Type of databases are document based, key-value pairs, graph databases, wide-column stores or XML databases.	Table based databases which consist of rows of data.
Uses dynamic schema for unstructured data.	Uses redefined schema.
Horizontally scalable, needs more members in the pool.	Vertically scalable, needs more volume per machine.
Unstructured Query Language.	Structured query language.
Preferred for large data set.	Better for complex queries.
MongoDB, BigTable, Redis, RavenDb, Cassandra, HBase, Neo4j, CouchDb etc.	MySql, Oracle, Sqlite, Postgres, MSSQL etc.

NoSQL databases are better to scale and for tuning their performance. It can be done by adding more members to a pool, not by changing hardware per machine. Adding hardware per physical machine cannot be done without downtime, which means NoSQL databases are more flexible and are preferred with a large data set.

Top document-based NoSQL databases are MarkLogic, MongoDB, Apache CouchDB, OrientDB, IBM Cloudant, BaseX [23]. A standard security implementation will be investigated between MongoDB, OrientDB and Apache CouchDB to take it into account in securing Elasticsearch. Every system should have a proper authentication and authorization process. Before going further with NoSQL security, authentication and authorization processes will be explained.

4.2.1 Authentication vs authorization

Authentication refers to the process, where someone or something will be identified. Authentication technology provides access control for systems to check if a user's credentials match with the one that is stored in the database [24]. Users will be identified with a user ID and the authentication process will succeed if a user provides information that a concrete user should only know. It is an authentication factor, which is credential that matches with a user ID in the database, for example a password [24]. It is important to provide information only to authorized people to keep networks secure [24].

Authorization is a process where a permission to do or have something will be granted after authentication. System administrators can define for the system, who can access the system and share privileges to users [25]. System administrators must have a control of their system and how much permissions users have. Users need to have only the amount of access that is essential to their work and no more [26].

In the next paragraph an overview of NoSQL database security implementations will be given.

4.2.2 NoSQL database security (MongoDB, OrientDB, Apache CouchDB, Elasticsearch)

In this paragraph the author gives an overview on how different NoSQL databases implement their security. The author takes useful tips from them to raise requirements for securing Elasticsearch.

MongoDB

Each application and user of a MongoDB system should map to an exact user. This access isolation simplifies access revocation and continuing user maintenance [27].

User need to have a username, password and authentication database associated with that user [27].

Fresh install from the localhost allows to enable access control [28]. This means that user can connect to the database simply by telling MongoDB in the Linux shell and then

the admin access will be given. After that it is possible to create a first user in the system [28].

Starting from the MongoDB 3.4, it is possible to create users in MongoDB via LDAP [Lightweight Directory Access Protocol] authorization. It is possible to create a role inside of MongoDB that maps to a role defined in LDAP. This means that all MongoDB users must be stored in LDAP server [28].

After creating a first user with credentials, a localhost authentication without a password needs to be disabled to avoid unauthorized access to the database.

MongoDB uses RBAC to determine access for users. A user is granted one or more roles that allows the use of MongoDB resources and the action that the user can do [29].

Default authentication mechanism is SCRAM [Salter Challenge Response Authentication Mechanism] in MongoDB [30].

SCRAM provides tunable work factors, per-user random salts and the authentication of the server to the client as well as the client to the server using SCRAM-SHA-1 or SCRAM-SHA-256. MongoDB informs that the iteration count must be 5000 or greater [31].

TLS/SSL connection is supported for the client authentication and the internal authentication of the members of replica sets in MongoDB and sharded clusters with a minimum of 128-bit key length for all connections [32], [33]. But with Linux legacy x64 builds of MongoDB, TLS/SSL connection is not possible [33].

LDAP proxy authentication is supported by MongoDB Enerprise. It supports querying an LDAP server for the LDAP groups the authenticated user is a member of. MongoDB directs the DNs [Distinguished Name] of each group to roles on the admin database. The user will be authorized based on the mapped roles on the admin database and their associated privileges [34].

Every Active Directory object has a DN that identifies the object and describes the full path to an entry [35].

Kerberos authentication is supported in MongoDB Enterprise. Every member in the authenticated communication in a Kerberos-based system is known as a “principal” and it must have a unique name. Principals are in administrative units, which are known as realms. The KDC [Kerberos Key Distribution Center] keeps a database of each realm’s principal and the principals’ associated “secret keys”. For a client-server authentication, the client requests a “ticket” from the KDC for access to a concrete asset. KDC uses the client’s secret and the server’s secret to make the ticket. This allows the client and server to mutually authenticate each other not knowing the secrets. In MongoDB there are user principals and service principals. Service principal authentication keys are stored in keytab files. It should have access only to user that runs the mongod or mongos process. On Linux, clients can use the Kerberos’s kinit program to authenticate the user principal to servers [36].

With internal authentication, it is possible to authenticate members of replica sets and sharded clusters to each other. One of the ways is to use keyfiles which use SCRAM mechanism, where keyfiles contain a shared password for the members. The key must be between 6 and 1024 characters in base64. Another way is to use x.509 certificates for replicas and sharded clusters for internal authentication [37].

Usually clients authenticate directly to mongos instances in sharded clusters. The mongos instances forward queries and write operations to the shards, but some maintenance operations may require authenticating directly to a shard [27].

OrientDB

OrientDB uses the greatest focus on security between NoSQL databases. A user has to authenticate to connect to an existing database. After starting service for the first time, it creates a root account automatically and asks to give password to it, and when the user does not provide it, the database gives a random password. OrientDB uses PBKDF2WithHmacSHA256 hashing algorithm for passwords if it is running at least on Java 8, otherwise PBKDF2WithHmacSHA1 with configurable Salt [38]. It uses 24-bit length Salt per user for a configurable number of iterations, by default is 65 536 iterations. The higher iteration count can slow down attacks, but also slows down the authentication process. To speed up password hashing, OrientDB uses a configurable password cache. Although, it is possible to switch off the cache [39]. CPU intensive

algorithms such as PBKDF2, Bcrypt or Scrypt take a work factor or iteration count as an argument to make hash function slower and to add additional security to passwords. Using the PBKDF2 algorithm in password hash, it makes brute force attacks slower [40].

OrientDB enables to create different roles and modify those rules to make exceptions. Roles can have operations like create, read, update, delete, which can be combined in between them or all operations in one role together. Moreover, OrientDB allows to manage security per record [39].

Furthermore, OrientDB enables to encrypt records on a disk. The encryption key must be provided at run-time, the key would not be saved on the database. A database encryption can only be used on local database [41]. It supports AES [Advanced Encryption Standard] and DES [Data Encryption Standard] algorithms for encryption. AES algorithm is recommended, because it is stronger. OrientDB can have multiple databases with different encryption interfaces, running under the same server [41].

Moreover, OrientDB has support for its HTTP [Hypertext Transfer Protocol] and BINARY protocols through SSL [42].

Apache CouchDB

Apache CouchDB allows any query to be made without an authentication and listens on loopback network interface by default [43]. Everyone is an admin user by default with server admin credentials, but it is possible to create admin users with a username and password as their credentials [43]. To change default permissions, a `_security` document needs to be created [43].

Admin users can do anything in CouchDB. Admins can read and write all types of documents, modify which users are members or admins and set per-database configuration options, while members in CouchDB can read all documents and create and modify any document except design documents [43]. To make specific users to only have a read-only opportunity, every database in CouchDB must have a separate design document with validation functions, which makes sure if a user is able to modify documents or not. One database can have multiple design documents. If validation

functions have not been used, then no checking is done and documents with any content or structure can be written into CouchDB database [44].

Passwords in CouchDB are hashed in normal operation when the log level is not set to debug mode [43]. CouchDB creates 128-bit UUID [Universally Unique Identifier] Salt which is added to a plain-text password and then it creates a SHA1 hashed password of it [43]. CouchDB 1.1.0 has a SSL connection built in [43].

SHA1 hashes are not the strongest ones. With reference to an article in February of 2017: “For more than six years, the SHA1 cryptographic hash function underpinning Internet security has been at death's door. Now it is officially dead, thanks to the submission of the first known instance of a fatal exploit known as a "collision"” [45]. If attacker knows, how the Salt is made, it is possible to guess SHA1 password to get plain-text passwords in CouchDB database. Adding Salt to password mitigates password attacks like rainbow tables [46]. According to OWASP, “Salt is a unique, randomly generated string that is added to each password as part of the hashing process” [47]. This means that attacker needs to crack hashes one at a time, because the Salt is unique for every user [47].

CouchDB has also cookie authentication, which means that it will generate a short-term token that client can use in its next request to CouchDB. A token is valid for 10 minutes by default [43]. This means that access token is linked with a user in a CouchDB database and user does not have to use username and password for every request.

Elasticsearch security

Elasticsearch enables to authenticate with Active Directory, LDAP or the Elasticsearch native realm. Authentication is possible with certificates, Kerberos and SAML. Different user and role permissions enable to make specific rules, for example so that IT team can monitor Elasticsearch health without being able to see or modify the data. Also, access to specific indices in Elasticsearch is possible [48].

In Kibana, it is possible to give permissions to specific dashboards, visualizations [48].

SSL/TLS encryption enable to secure node-to-node traffic, HTTP requests and transport client traffic across Elastic Stack. Moreover, IP filtering can prevent unapproved hosts from joining or communicating with Elasticsearch cluster [48].

Many Elasticsearch features for security are not freely available by Elastic. Free version of Elasticsearch is only in basic license, that includes some of the X-Pack security features. The basic license includes security features like TLS, file-based authentication and role- based access [49]. Elastic added RBAC to its basic license starting from version 7.1 [50]. This means that before version 7.1, Elastic had authentication only with paid versions. Furthermore, there is elasticsearch-oss, which enable to collect logs, search logs, create dashboards and clusters, but with oss you cannot use TLS and role-based access control, which are available in basic licence that is also free version of elasticsearch [51]. Audit logging, IP filtering, LDAP, single sign-on options, field- and document-level security, encryption at rest support is only available with eneterprise version. Moreover, basic licence does not include alerting and machine learning features like anomaly detection [51].

Elasticsearch uses REST API [Application Programming Interface] to configure and access Elasticsearch features directly by the UI [User Interface] components [52].

Table 2 gives an overview of features that are included in Elasticsearch versions and is based on the information found in official Elastic page [51]. "+" sign means this is included feature and "– " sign shows this is not included with that Elasticsearch version. Free versions are open source and basic license. Gold, platinum and enterprise versions cost money. Different functionalities based on Elasticsearch licenses are outlined in Table 2. Descriptions of features are explained after the Table 2.

Table 2. Functionalities based on the Elasticsearch licenses.

Functionality	**Open Source (OSS)**	**Basic license**	**Gold license**	**Platinum license**	**Enterprise license**
Snapshot/restore	+	+	+	+	+
Minimal snapshots	-	+	+	+	+
Clustering & high availability	+	+	+	+	+
Cross-cluster search	+	+	+	+	+

Functionality	**Open Source (OSS)**	**Basic license**	**Gold license**	**Platinum license**	**Enterprise license**
Voting-only master nodes	-	+	+	+	+
Encrypted communications	-	+	+	+	+
Role-based access control	-	+	+	+	+
File and native authentication	-	+	+	+	+
Kibana spaces and feature controls	-	+	+	+	+
API keys management	-	+	+	+	+
Audit logging	-	-	+	+	+
IP filtering	-	-	+	+	+
LDAP, PKI, Active Directory authentication, including attribute-based access control	-	-	+	+	+
Single sign-on (SAML, OpenID Connect, Kerberos)	-	-	-	+	+

Functionality	Open Source (OSS)	Basic license	Gold license	Platinum license	Enterprise license
Field- and document-level security	-	-	-	+	+
Encryption at rest support	-	-	-	+	+
Full stack monitoring	-	+	+	+	+
Alerting	-	-	+	+	+
Anomaly detection	-	-	-	+	+

Snapshot is for making backups of the Elasticsearch cluster. It can be made of individual indices or entire cluster. Snapshots are incremental, which means that an index of a snapshot stores only data that is not part of an earlier snapshot. Restoring can be made through restore API [53].

Minimal snapshot is a source-only snapshots that contain stored fields and index metadata. Index or doc values structures and not included and are not searchable when restored. Source-only snapshots need reindexing the data into a new index after restoring [54].

Clustering and high availability using replica shards provide availability of a service if accidental loss of a node happens or if operating system patching takes place [55].

Cross-cluster search enables to run a single search request against one or more remote clusters [56].

Voting-only master node is a node that participates in master elections but will not act as cluster's elected master node. It only chooses the next master node, but itself would not be a master node. High availability clusters require at least three master-eligible nodes and at least two of them are not voting-only nodes [57].

Encrypted communications mean that encrypted traffic into, from and within cluster can be configured using TLS/SSL connection [58].

Role-based access control provide a way to add several users and restrict their access to specific platform resources [59].

File and native authentication, file authentication can be made with file realm, which are defined in local files on each node in the cluster. It is useful as fallback realm, for example if users have been locked themselves out of the system [60]. Native authentication is with the internal native realm, that can be used through REST APIs or Kibana to add and remove users, assign user roles and manage user passwords. By default, user credentials are stored in memory using a salted SHA-256 hash algorithm and a hashed version of passwords are stored on disk salted and hashed with bcrypt hash algoritm. Morover, other hash algoritms can be configured to be used [61].

Kibana spaces and feature controls can be used for securing Kibana's dashboards and saved objects. This means that Kibana allows to organize dashboards, visualizations and other saved objects into different categories. Role-based access control can be used to define who can view and edit which space [62]. Furthermore, it is possible to have control which features are available to which users in Kibana [63].

API keys management can be done from the Kibana Console. API keys enable to create secondary credentials so that requests can be send on behalf of the user, for example using them in a cron job [64].

Audit logging is for monitoring authentication failures and refused connections in Elasticsearch cluster for suspicious activity [65].

IP filtering can be applied to application clients, node clients or transport clients in addition to other nodes that are attempting to join the cluster. This enables to put IP addresses into blacklist and their connection will be dropped if they try to connect [66].

LDAP, PKI, Active Directory authentication, LDAP can be used to communicate with Active Directory server, so that active_directory realms will be used to authenticate users. Moreover, active_directory realm enables to map Active Directory users to roles via Active Directory groups or other metadata [67]. Furthermore, PKI certificates can be used to authenticate users. This means that clients connect directly to Elasticsearch to present X.509 certificates. The certificates must be first accepted for authentication on the SSL/TLS layer on Elasticsearch and then validated by PKI realm [68].

Single sign-on can be configured into Kibana, using Elasticsearch as a backend service [69].

Field- and document-level security, this security implementation can restrict the documents and fields that users have read access to [70].

Encryption at rest support, all the data that is stored can be encrypted, which is filesystem encryption. Elasticsearch team will support encrypted filesystem environments with Platinum license, otherwise they need to reproduce the problem outside of an encrypted filesystem environment. It is not a feature of Elasticsearch itself [71].

Full stack monitoring, this gives an overview of operation of Elasticsearch, Logstash and Kibana. All metrics are stored in Elasticsearch and can be viewed from Kibana [72].

Alerting, this informs about changes in data and alerts can be sent for example to email, Slack or to PagerDuty. All alert executions are indexed into Elasticsearch, that can be seen from Kibana [73].

Anomaly detection automatically models the normal behaviour of time series data to identify anomalies in real time and to accelerate root cause analysis. It has also machine learning for creating anomaly detection jobs and understanding results [74].

In the next paragraph, an overview of the centralized logging system of Elastic Stack will be given starting of Elasticsearch use cases.

4.2.3 Elasticsearch use cases

Elastic products are used by ecommerce websites like eBay, social media platforms like Facebook, media search platforms like BBC, health services like Influence Health, taxy service provider Uber and some firms uses it to monitor malicious activities with Elasticsearch, for example Slack, Sunhotels and Oak Ridge National Laboratory [75].

Uber uses it for their core data system to collect metrics in real time. They collect different metrics about pricing, supply positioning, and assess overall marketplace diagnostics [76].

eBay is used by millions of sellers, has 162 million active buyers and 800 million listings. They use Elasticsearch to handle all their search functionalities across the business with 18-19 clusters [77].

Facebooks uses it to search over 40 tools across multiple clusters with more than 60 million queries a day [78].

Netflix uses Elasticsearch for their messaging system. It includes emails, app push notifications and text messages. In Kibana, they can see in real time how many people got notified with the message, the message delivery success rate and why some of the messages have failed [79].

HG Insights, that collect sales and marketing intelligence data [80], tracks that Elasticsearch is used by 36 193 companies and Kibana by 12 467 different firms [81].

Companies choose Elasticsearch, because it can be used in various ways and is scalable solution by adding more nodes to perform better. It can be used to store main data, create searchable catalogue, document store and logging system. Also, for complementary technology to add visualization capabilities to SQL, MongoDB. Furthermore, it enables to add metrics, monitoring and analytics capabilities [79].

Elasticsearch allows to use aggregation to get big picture of data and it can combine different type of searches: structured, unstructured, Geo, application search, security analytics, metrics, and logging [79].

4.2.4 Elastic stack

This paragraph gives an overview of Elastic products in centralized logging system in more detail.

Elastic stack has grown out of the ELK stack. ELK stack includes Elasticsearch, Logstash and Kibana. Elastic stack includes also Beats [82]. Figure 1 [83] gives an general overview of Elastic stack.

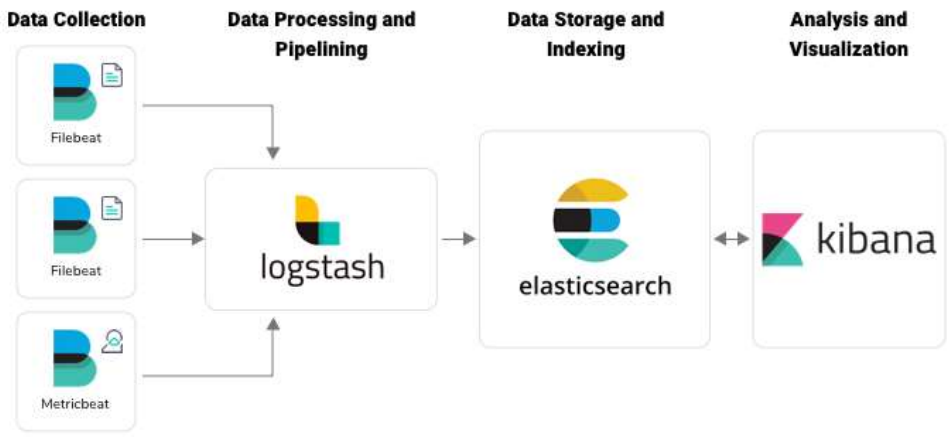

Figure 1. Elastic Stack.

Beats are a data senders from hosts to Elasticsearch and are placed into a host. It can send data to Logstash or Elasticsearch directly. Elastic has different kind of data shippers. Filebeat for logfiles, Metricbeat for system and services metrics like CPU and memory usage, packetbeat for network traffic, winlogbeat for Windows event logs, auditbeat for audit framework data, heartbeat for monitoring service uptime and functionbeat for cloud data to collect, ship and monitor data from cloud services [84].

Logstash is a Elasticsearch component on the server-side and is used for the raw logs to process them and ingest data from multiple sources simultaneously to Elasticsearch. Moreover, it transforms and filters logs and then sends it to Elasticsearch in the form of index [82].

Elasticsearch is a full-text search and analytics engine that enables to store, search and analyse big amount of data in the form of indexes. It is used by applications that have complex search features and requirements [85]. All the data in the Elasticsearch is indexed and shared into shards between Elasticsearch nodes.

Kibana is graphical interface of Elasticsearch that enables users to visualize data with charts and graphs [82].

Elasticsearch cluster is collection of collected nodes [57]**.** Figure 2 [86] gives an overview of Elasticsearch cluster with two nodes and how they connect to an Elasticsearch index.

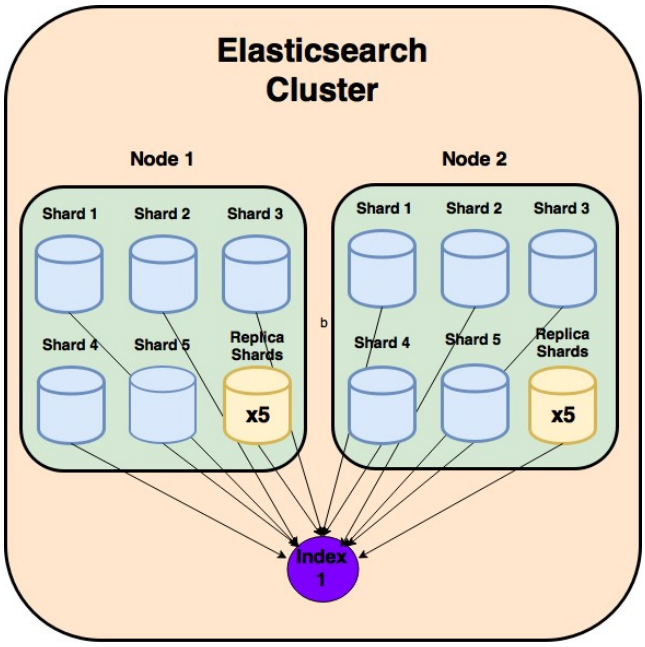

Figure 2. Elasticsearch Cluster.

As soon as Elasticsearch is installed and started in a server, it is starting a **node**. If Elasticsearch runs on a single node of Elasticsearch, then it is an Elasticsearch cluster of one node [57].

Index is like a database in a relational database that has mapping with multiple types. It is a logical namespace which maps to one or more primary shards and can have zero or multiple replica shards [87].

Shard stores Elasticsearch index functionality [87]. Shards are divided between multiple nodes and one index can share its' data between multiple shards. This enables to scale content volume per node and increase Elasticsearch performance. In case if shard goes offline and data would be accessed without a loss, shards can have copies of a shard on a different node, **replicas**. Moreover, it provides to scale out Elasticsearch search throughput, as searches can be executed on all replicas in parallel [88].

4.2.5 Alternatives for Elastic Stack

One of the alternatives is Apache Solr, that is based on the Lucene search library as Elasticsearch with XML/HTTP and JSON APIs. Solr provides full-text search, hit highlighting, faceted search, dynamic clustering, database integration. It provides distributed search and index replication and is scalable as Elasticsearch [89]. Solr does not have query analyser chain support for nested documents and support for multiple

document types per schema. Furthermore, it does not have ability to re-locate shards and replicas, also to change a schema without restarting the server and search across multiple indexes [90]. Lucene itself that Elasticsearch uses for its indexing and search, is more complex and does not have distributed real-time engine [91].

ExpertRec enables to build search engine from scratch and edit every feature of it [91].

Another alternative can be Graylog, that is also open source logging software. Graylog cannot read from collected syslog files and needs sending logs directly from an application to Graylog [92] [93]. Its GUI [Graphical Unit Interface] allows to manage permissions [92]. Graylog enables to centralize and aggregate all log files and has powerful query to search through terabytes of log data and analyse information [94].

Logentries is for log data to make it easily accessible to IT operations, development and business analysis teams. It has broad platform support and open API [95]. It allows to extract field level values, analyse them with powerful search functions, and visualize them with detailed dashboards [96]. It has real-time search through indexed logs and across the Logentries system [97]. On the other hand, it is not open source and costs money. Only free trial is available [98].

Elastic Stack is a powerful set of opensource tools which can be set up quickly to have centralised logging server. The most similar to Elastic Stack for logs is Graylog, but Graylog receives logs directly from application through the network protocol, Elasticsearch can read previously collected plain text logs using Logstash and then parsed to Elasticsearch [93]. Elastic Stack has tools to transfer logs from hosts to centralized logging server and it can read raw log files. Elasticsearch stores information, enables data modification and has a visualization tool for querying data, dashboards and graphs. It can understand different formats of data, stores information, gives fast search results and enables to analyse its data. The author will continue implementing and improving Elastic Stack in her thesis after getting familiar with alternatives.

4.2.6 Elasticsearch data breaches

In this paragraph some of the biggest data leakage examples have been outlined that had been leaked from the unprotected Elasticsearch server.

On March 2020, 5 billion records were leaked from unprotected Elasticsearch server. It was discovered by Bob Diachenko, the leaked database contained security incident details from 2012 to 2019, like hashtypes, leak date, encrypted or plaintext passwords, emails, source of the leaks from the U.K-based security firm. The data included information about Adobe, Last.fm, Twitter, LinkedIN, Tumblr and other security incidents [99].

On February 2020, French sports store had data leakage in which more than 123 million records leaked from improperly secured Elasticsearch server. Customer emails and passwords, API logs, private information of employees, contract details, dates of birth were revealed [100].

On October 2019 leaked personal and social information of 1.2 billion people. 4 billion user accounts from unprotected Elasticsearch server that had no authentication implemented. The leaked data involved names, email addresses, phone numbers, LinkedIN and Facebook profile information. It contained data sets that originated from two different data enrichment companies [101].

In January 2020, was reported that 250 million Microsoft customer service and support records were exposed on the web. The data contained logs of conversations between Microsoft support agents and customers from over the world in a period of 2005 to December 2019. This contained email addresses, IP addresses, locations, descriptions of Customer Service and Support claims and cases, case numbers, resolutions, remarks, internal notes that were marked as confidential. The data was accessible to anyone with web browser, with no password or other authentication needed [102].

In July 2019, it was published that Elasticsearch data leakage happened in China. More than 90 million records leaked by Chinese public security department from publicly accessible and unsecured Elasticsearch server. The data that was leaked contained names, birth dates, genders, identity card numbers, location coordinates and city relations for individuals, business IDs, types, location coordinates and memos of the businesses [103].

In November 2018, it was published that nearly 57 million Americans personal information was leaked from unprotected Elasticsearch server. The data contained

personal information, like first name, last name, email address, home address, state, ZIP code, phone number, and IP address [104].

This paragraph gave examples, why Elasticsearch data should be protected from public networks and the need for securing Elasticsearch.

Elastic team suggests putting Elastic Stack behind a VPN or firewall to protect access from public networks to Elasticsearch data [105]. Elastic has advised to secure Elastic Stack by encrypting communications, role-based access control, IP filtering and auditing to set passwords for the server's built-in users, and to properly configure the instance before deploying it in production [103].

4.3 Requirements for securing Elasticsearch

In this paragraph, the author states the requirements for securing Elasticsearch.

Elasticsearch basic license does not have audit logging, LDAP integration and alerting possibility. Open Distro Security for Elasticsearch, Search Guard and ReadOnlyREST enable to have encrypted authentication and role-based access control to Elasticsearch. In this paragraph the author analysis these tools to see if there is a better alternative for securing Elasticsearch and that does not cost a lot of money.

The requirements are described in Table 3. The first column represents the number of a requirement, the second column is the description of the requirement and the third column is an importance category that can be high (mandatory), medium (recommended) or low (nice to have). The idea of evaluation to find the best solution is taken from this thesis [1] because it suited to my thesis to answer the problem and is adjusted to the needs of the concrete thesis.

Table 3. Requirements for Elasticsearch security tools.

Nr	Requirement	Importance
1	Works on Linux/Unix servers.	High
2	Audit logging for knowing who has been logged in or out and when or who has tried to log in but was not able to. It is important to find out who might have caused a security event, which could affect information security.	High
3	Full stack monitoring to have an overview of Elasticsearch, Logstash and Kibana's health.	High
4	Alerting to have notifications of changes in data that needs intervention.	High
5	Supports encrypted communications (TLS/SSL), to have additional layer of protection for information to avoid data leakage.	High
6	Role-based access control is important for choosing who can access what kind of information.	High
7	Kibana spaces and feature controls for choosing who can access what kind of dashboards and saved objects.	High
8	Uses Elasticsearch	High
9	Field and document level security to have possibility to choose who can read which documents and what kind of fields.	High

Nr	Requirement	Importance
10	Opensource solution, because this allows to view the source code by anyone and share knowledge to improve the product.	High
11	Has no existing vulnerabilities according to CVE [Common Vulnerabilities and Exposures] list.	High
12	Regular development of that product - at least every week should have new commits with improvements in the source code.	High
13	LDAP, Active Directory authentication is better for managing user permissions. It forces users to have complex passwords and it asks to change user password after a certain time period. Also, if people leave their job position, it is possible to quickly disable its' account from one place, so that he or she would not be able to log in and do something.	Medium
14	Available documentation of a tool	Medium
15	API interface	Low
16	Data import / export	Low
17	Customer support	Low
18	Forum	Low
19	Additional security features	Low
*	The cost	?

The purpose is to find the best security solution for Elasticsearch that could be found in the market. The best solution will be chosen with the highest score of relevant requirements. Mandatory requirements will give 3 points, medium requirements will give 2 points and low ones 1 point for each feature. If the feature is not found from the materials, then it is marked as unknown and counted with 0.

*The cost will be considered if the final score is the same between the products. The product with a lower cost will be chosen.

4.3.1 Open Distro Security for Elasticsearch

Open Distro for Elasticsearch is an Apache 2.0-licensed distribution with enterprise security features. It includes alerting, SQL and cluster diagnostics [106].

It enables Active Directory Authentication, Kerberos or JSON web tokens for single sign-on. Also, it monitors and logs any malicious access attempts. It supports OpenSSL and TLS 1.2 connection. Moreover, it has role-based access control to control cluster operations, access to indices, fields and documents. Open Distro allows multiple teams to share the same cluster while being able to access specific data and dashboards [107].

Open Distro uses Kibana interface and API. Monitors can be set up with visual editor or with an Elasticsearch query. What is more, it can have multiple trigger conditions for generating alerts. Alerts can be integrated with webhook and alerts can be formatted with Mustache to embed relevant information in the notification. All alert executions are indexed in Elasticsearch that can be tracked and seen in Kibana [108].

What is more, Open Distro enables to query data with SQL language and provides data export to CSV. Data can be read in JSON documents or CSV tables. SQL statements can be translated into Elasticsearch JSON queries with `_explain` call, so that SQL can be used in making JSON queries to configure access control policies in Security and monitors in Alerting. Moreover, Open Distro for Elasticsearch provides JDBC driver for integrations with business intelligence, analytics and ETL tools to extract and analyse Elasticsearch data [109].

Performance Analyzer uses REST API for querying Elasticsearch metrics, like consumption of network, disk, and operating system resources. This runs separately of the Elasticsearch cluster to diagnose and resolve issues, even the cluster is not

performing. It can be used with a PerfTop CLI [Command Line Interface], that provides pre-configured dashboards for analysing cluster, node and shard performance [110].

Open Distro for Elasticsearch monitors indices and has index state management policies that can be applied on routine tasks to index patterns and move indices from one state to another. In policy stages, users can define criteria like index age, size or number of documents and define actions like force merge, snapshot, rollover when transitioning to a new state [111].

In the future, Open Distro for Elasticsearch will add its' anomaly detection feature that uses Random Cut Forest machine learning algorithm, which is an unsupervised machine learning algorithm that computes an anomaly score for each incoming data point. "The final anomaly score is an average score of all data points and used to differentiate an anomaly from normal variations" [112]. Moreover, Open Distro for Elasticsearch will add SQL console to Kibana, which makes running SQL queries easier and exploring Elasticsearch data [112].

Data export and import can be done through API requests.

4.3.2 Search Guard

Search Guard is a security plug-in with alerting opportunity for the Elastic Stack that encrypts and protects its' data and data flows. It uses role-based access control on clusters, indices, fields and documents with encrypted communication and authentication, authorization and audit logging. Moreover, it supports LDAP authentication and authorization, OpenID, SAML, Kerberos, JSON web tokens and client certificates [113].

Serarch Guard has active community and customer support. They have priority support for customers with a 48h response time [113].

There are several search guard versions.

Standard edition enables to choose from community edition, which is free of charge and has basic security needs; then Enterprise edition that can be integrated with Active Directory and has field and document-level security and access control and management

API; then compliance edition that is designed for meeting compliance regulations like GDPR and records all read-and-write access to data and monitors integrity of Elasticsearch installation [113].

Community edition has REST encryption, inter-node encryption, PEM and OpenSSL support, role-based access control, transport- and HTTP access control, cluster-level and index-level access control, internal user management, HTTP Basic authentication, PKI and proxy authentication and internal group management. Moreover, alerting possibility, Kibana access control, SIEM compatibility and Elastic Stack Monitoring and Elastic Stack Machine Learning opportunity [114].

Enterprise version adds document-level and field-level access control to Elasticsearch and authentication can be used with LDAP, Kerberos, JSON web tokens, OpenID Connect, SAML or with custom implementations. Enterprise version can send alerts to Jira and PagerDuty action. Its' configuration can be made through REST management API or from configuration GUI and Kibana can be used with Single Sign On [114].

Compliance edition adds field anonymization and immutable indices to Elasticsearch. It has more specific audit logging that includes security audit logging, read-access, write-access audit logging, tracks configuration changes and system changes [114].

Academic and custom versions are Academic and Scientific edition and custom edition [113].

Academic and Scientific edition includes Search Guard Enterprise and Compliance licensed free of charge and is for all non-profit academic and scientific projects. Can be used on unlimited clusters and nodes [113].

Custom edition is for OEM [Original Equipment Manufacturer] partners and system integrators and has direct support from the Search Guard team [113].

Enterprise version with trial was chosen to comparison, because it has LDAP authentication possibility and document- and field-level access control.

4.3.3 ReadonlyREST

ReadonlyREST free version is plugin for Elasticsearch that enables TLS encryption, customizable audit logging and internal or external authentication and authorization [115]. What is more, it is using ACL [Access Control List] [115].

ACL can be used as IP filtering option to choose, which IP ranges or hostnames can connect to Elasticsearch cluster and which indices [115].

It uses HTTP Basic Authentication, JSON Web Tokens and LDAP for authentication and has internal user groups, external Json Microservice availability and LDAP Authorization connector for authorization [115].

It has network level, index, field and document level access control [116].

Free version does not contain multi tenancy for different Kibana dashboards, different roles to Kibana features and settings, SAML authentication, authorization and customer support. It is included in Pro and Enterprise licenses [116]. Multi Factor Authentication is provided by their SAML SSO connector which can be integrated to LDAP server via solutions like Microsoft Active Directory Federation Services [117].

4.3.4 Apache authentication

Access to Elasticsearch from Kibana can be protected with Apache authentication. This means that Apache webserver can be configured to make authentication by itself. There are two types of authentication in Apache basic and digest.

Apache basic authentication needs more configuration to make it secure. As a default, it sends the password from the client to the server unencrypted, unless using it with module mod_ssl [118]. Using this module with require directive, the ssl provider will deny access to server if a connection is not encrypted with SSL [119].

Apache HTTP digest authentication does not transmit password as cleartext. Its' password storage on the server is less secure with comparison to basic authentication. Better alternative is to use basic authentication with mod_ssl for encrypting the connection [120].

Apache uses AuthBasicProvider to verify the user and if the user is not listed in any its' provider then access will be denied. Providers are performed by mod_authn_dbm, mod_authn_file, mod_authn_dbd, mod_authnz_ldap and mod_authn_socache [121].

Mod_authn_dbm works with basic and digest authentication and searches users in dbm password files [122].

Mod_authn_file works with basic and digest authentication and searches users from plain text password files [123].

Mod_authn_dbd works with basic and digest authentication and searches users in SQL tables [124]. Digest authentication can understand only MD5 hash of the string. Basic authentication can understand bcrypt, MD5, SHA1 and CRYPT algoritms [125]. SHA1 and MD5 are older hashing algorithms that have cryptographic weaknesses and are not trusted by modern browsers, newer hashing algorithm SHA256 should be preferred [126]. As written before, bcrypt uses iteration count to make hashing function slower and to make password guessing harder. Crypt is a Linux password and data encryption function with variations [127].

Mod_authnz_ldap module is for the Apache basic authentication and it enables to authenticate users through LDAP directory. It uses two phases in granting access to user. In the first phase authentication provider verifies if the user's credentials are valid, also called the search/base phase and in the second phase mod_authnz_ldap figures if the authenticated user can access the resource, known as compare phase [128].

Mod authn_socache uses a cache of authentication credentials to relieve the load on backends, so that new backend lookup is not required for every authenticated request. SQL based authentication provider mod_authn_dbd can benefit from this cache [129].

In addition, Apache can be configured to use client authentication with certificates [130]. This allows the use of smartcards in authentication process. For example, using Estonian EID smartcards in Ubuntu Apache2 webserver [131]. Apache allows to choose cipher suits in authentication process [131]. This enables to choose stronger ciphers for authentication to make it more secure. According to OWASP cheat sheet, null ciphers, anonymous ciphers and EXPORT ciphers should always be disabled. If possible, then only GCM ciphers should be enabled [126].

Commercial support is available from third-party companies [132].

4.3.5 Comparison of Open Distro Security for Elasticsearch, Search Guard, ReadOnlyREST, Apache authentication and Elasticsearch basic license

The comparison will be made according to the requirements that have been outlined in the Table 3. "+" means that it has the requirement included, "-" marks that the requirement is not included. The author has marked value as unknown, if the feature was not found in the materials and will be clarified in the testing. Comparison between chosen products has been demonstrated in Table 4.

Table 4. Comparison between security tools for Elasticsearch.

Requirement	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Readonly-REST free version**	**Elastic-search basic license**	**Apache authentication**
Works on Linux/Unix servers	+	+	+	+	+
Audit logging	+	+	+	-	-
Full stack monitoring	+	+	-	+	-
Alerting	+	+	-	-	-
Supports encrypted communications (TLS/SSL)	+	+	+	+	+
Role-based access	+	+	+	+	-

Requirement	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Readonly-REST free version**	**Elastic-search basic license**	**Apache authentication**
Kibana spaces and feature controls	+	?	-	+	-
Uses Elasticsearch	+	+	+	+	-
Field and document level security	+	+	+	-	-
Opensource project	+	+	+	+	+
No existing vulnerabilities according to CVE list	+	+ [133]	+	+ [134]	- [135], [136]
Regular development of that product	+ [137]	+ [138]	+ [139]	+ [140]	+ [141]
LDAP, Active Directory authenticat-ion, SAML	+	+	LDAP exists, SAML with commercial license	-	+

Requirement	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Readonly-REST free version**	**Elastic-search basic license**	**Apache authentication**
Available documentation of a tool	+ [142]	+ [143]	+ [144]	+ [145]	+ [146]
Uses API interface	+	+	+ [115]	+	-
Data import/export	+	Unknown	Unknown	Unknown	-
Customer support	-	+	-	-	+ [132], [147]
Forum	+ [148]	+ [149]	+	+ [150]	+ [147]
Additional features	SQL queries, JDBC driver	Elastic Stack Machine Learning opportunity	Network access control	API keys	-
Full score	44 points	41 points	33 points	32 points	18 points
The cost	No additional cost	2700 EUR per node annually	No additional cost with free version	Basic version has no cost	No additional cost

Elasticsearch versions before 6.8.8 and 7.6.2 have privilege escalation flaw if generating API keys is possible [134].

Search Guard Kibana Plugin has vulnerable versions before 5.6.8-7 and before 6.x.y-12, which had an issue that an authenticated Kibana user could impersonate as kibanaserver [133].

In Apache HTTP server 2.4.0 to 2.4.41, using mod_rewrite, it is possible to redirect instead to an unexpected URL [Uniform Resource Locator] within the request URL [135].

"In Apache HTTP Server 2.4.0 to 2.4.41, mod_proxy_ftp may use uninitialized memory when proxying to a malicious FTP server" [136].

As a summary, Search Guard is a great tool for Elasticsearch, but community edition can be paralleled to Elasticsearch licenses itself. LDAP version is not available without a fee. Only advantage over Elasticsearch basic version is that it has alerting and Machine Learning opportunity with community edition. Free of charge version uses HTTP encrypted authentication as Apache but adds role-based access control to Kibana.

The author asked vendors about the prices, because in their webpage they did not showed the cost of solutions. Elastic gold package that includes LDAP authentication costs 12600 per year, whereas minimum purchase is for three nodes. Search guard enterprise version costs 15000 EUR per year for unlimited number of nodes and costs 2700 EUR per one node. 2700 x 3 = 8100 EUR. This means that Search Guard is cheaper if LDAP functionality is needed for three nodes. Moreover, if single sign on solution is needed for Kibana, then Elastic Platinum license must be chosen, and it costs 18900 EUR annually. Search Guard Enterprise version includes it in the same Enterprise package.

To answer the question marks that are outlined in a Table 4, features need to be tested practically, because all the information could not be found from the Internet. Moreover, every feature and implementation itself need to be tested to examine the complexity of a tool to make the final decision. The author chooses Open Distro Security for Elasticsearch and Search Guard Enterprise version for testing with biggest total score and Elasticsearch basic version for comparison in between of them.

The author installs Elasticsearch 6.8 for Open Distro Security and for Search Guard Enterprise, because it is still supported Elasticsearch version according to Elastic product lifecycle and it does not have authentication frontend as a free available option by Elastic with Elasticsearch 6.8. Elasticsearch 6.8 EOL is 20.11.2020 [151]. Moreover, after installing Elasticsearch 6.8, the author can try an upgrade of those solutions. Also, author chose Elastic stack 7.6 to be tested with basic license, which is the latest version of Elasticsearch at the moment.

5 Tests

This chapter describes the testing environment and raises testing questions that are considered to compare solutions practically. Configurations are described in Appendixes.

5.1 Testing environment

Testing environment was built in VMware vSphere 6.7 on ESXi 6.0 server.

The author made 3 virtual machines on Ubuntu 18.04.4 LTS [Long Term Support] with 16GB disk space, 4GB RAM and 4 CPUs. Ubuntu 18.04.4 was chosen, because at that time when virtual machines were created, it was the latest LTS OS [Operating System] for Ubuntu and the author has used to work with Ubuntu. Ubuntu 18.04.4 is still supported OS with EOL [End Of Life] in April 2028 [152]. All virtual machines were made in the same subnet that was meant only for testing purposes. Every virtual machine has its unique name and IP address.

Virtual machines:

1) elk.test.uni: Elasticsearch 7.6.2 with its' basic license, Kibana 7.6.2 and Logstash 7.6.2.

2) elk-sgrd.test.uni: Search Guard 40.0.0 with Elasticsearch 7.6.2, Kibana 7.6.2 and Logstash 7.6.2. At first Elasticsearch 6.8 was installed.

3) elk-opd.test.uni: Open Distro Security for Elasticsearch and Kibana 1.6.0 with Elasticsearch-OSS 7.6.1. At first Elasticsearch-OSS 6.8 was installed.

These virtual machines are suitable only for testing purposes. Usually production environment needs more volume, memory and CPUs. All virtual machines are integrated with Zabbix 4.4.6 to monitor virtual machines.

The author uses demo certificates to test the plugins and its features. These certificates are not recommended to use in production environment. All certificates must be changed if moving to production environment.

5.2 Testing questions

In this paragraph the author states the questions which are needed for choosing the best solution. Questions can be answered through testing security plugins.

- How much time did it take to install the plugin?
- How difficult was the installation?
- How easy it is to use the plugin? How difficult it is to add new users and groups? How to change default passwords?
- Does it have Kibana spaces and feature controls?
- Does it have data import/export possibility?
- Does the plugin do what it is supposed to do? Checking if it has features that are described in Table 4.
- Where are passwords stored? What kind of hashing algorithm is used?
- How does the plugin affect the performance of the test machine?
- How good was the documentation for the plugin?
- Is there something different from other plugins? Is it better or worse?
- How difficult it is to upgrade that plugin?

5.3 Testing Open Distro Security, Search Guard, Elasticsearch 7.6 with X-Pack

In this paragraph, comparison between Open Distro Security, Search Guard and Elasticsearch with X-pack functionalities will be given according to testing questions.

5.3.1 Open Distro Security

Admin and Kibana user password must be changed from the command line. The rest of the users can be changed in Kibana. If the system user default passwords has been changed, then all configuration files need also to be changed. The procedure has been described in Appendix 2.

The author did not choose to use Elasticsearch OSS 6.8, which is still supported Elasticsearch version, because Open Distro Security 0.10.0 which is compatible with that version was not available anymore. The author had to choose newer version. At first, the author tried to install Open Distro Security for Elasticsearch OSS 7.6.2 which is the newest, but it did not work with the latest Open Distro Security version 1.6.0. Elasticsearch OSS 7.6.1 had to be installed.

Compatible Open Distro versions with Elasticsearch can be checked from its webpage. [153]

Open Distro Security has very good documentation, where all the installation steps are logically aligned and the menu is understandable.

It took about one day to install Kibana, Elasticsearch and configuring using certificates. Installation is described in Appendix 2. Every minor version of Elasticsearch needs to be checked, if it is compatible with Open Distro Security service. Installation is smooth, if system has enough memory for installing Elasticsearch. It would not work if only 1GB of RAM is available. The author used sample data of web server logs which was previously installed.

Internal users are all hashed in internal_users.yml and passwords are stored in `.opendistro_security` Elasticsearch index [154]. The author did not find what was the default hashing algorithm that hasher uses for basic authentication for internal users. If using JSON web tokens in authentication, then it is possible to choose from various algorithms, for example HMAC-SHA256, HMAC-SHA512, ECDSA using P-512 and SHA-512 and all other standard algorithms [155].Only Kibana and Elasticsearch configuration files are using plaintext passwords.

Users can be added to internal database. Authentication backend can be basic HTTP authentication with noop, internal database or with ldap. Another way is to use Kerberos

authentication, JSON web tokens or via TLS client certificate [155]. Permissions can be granted to specific indices. Authentication backend must be configured in the backend server from the configuration file.

Users can be added to action groups and permissions (read, write, search, get, create etc.) can be shared to clusters, indexes, dashboards specifically.

To create a new user with minimal access rights, role mapping must be created. User into which action group and tenant has to be assigned in Open Distro Security. Tenants enable to have different Kibana spaces for user groups. Firstly, creating role name, action group from cluster permissions, where multiple action roles can be chosen. Then it is possible to choose tenant permission that apply globally, for example kibana_read_all. After making a role with all the permissions, then new user under role mapping must be created and assigned to that role.

Furthermore, Open Distro Security REST API enables to programmatically create and manage users, roles, role mapping, action groups, and tenants [156]. Account details can be looked from Kibanas' Dev Tools feature using API request with GET like this:

```
GET _opendistro/_security/api/account.
```

PUT request modifies details of user if it is used, for example changing user password:

```
PUT _opendistro/_security/api/account

{

    "current_password" : "old-password",

    "password" : "new-password"

} [156].
```

Possible API methods are: GET, PUT, POST, DELETE, PATCH [156].

Saved objects in Open Distro Security can be imported or exported, which includes saved searches, dashboards, queries, visualizations, configurations.

Load average is 0.0875 per core in 1 min with 4CPUs. Results have been outlined in comparison between tools in Figure 3 in the chapter 6. The author used data from Zabbix monitoring solution.

Audit logs are indexed per day to Elasticsearch and can be viewed from Kibana. Security-auditlog* contain 75 different fields including user, source, cluster name, node name, host name, host address, request privilege, remote address, privileges, keywords, category, timestamp, audit trace of indices, docs and so on.

Open Distro Security has also alerting possibility and index management through Kibana. In alerting, an index must be chosen which will be monitored and alerts can be send to Amazon Chime, Slack or custom webhooks.

Open Distro's performance analyser has to be installed separately which gives an overview of Elasticsearch metrics. Default version does not include it.

Open Distro packages can be used from repositories. To upgrade Open Distro Security, the first thing is to check if Open Distro version works with the latest Elasticsearch version. Then if Open Distro Security indexes need to be migrated to new version [157]. If Elasticsearch and Open Distro Security versions are compatible and Elasticsearch .deb package is upgraded, then Open Distro Security can be upgraded using `apt get install opendsirtoforelasticsearch` or `apt get install opendistroforelasticsearch-kibana` command in Debian Linux server as described in Appendix 2.

5.3.2 Search Guard

Search Guard configuration is kept in separate Elasticsearch indexes. All users, roles, permissions are in search guard indexes. Internal users' template is in file sg_internal_users.yml and uses bcrypt password hashing algorithm [158].

Search Guard authentication type can be defined in sg_config.yml. The procedure is defined in Appendix 3. Search Guard enables to define HTTP basic authentication (noop, internal, ldap), Kerberos, JSON web token, OpenID, SAML, proxy and clientcert authentication via a client TLS certificate. Noop if no other authentication against any backend system is performed [159].

The autc section in sg_config.yml is for configuring authentication to check credentials and authz for authorization, which defines how the roles for an authenticated user are retrieved and mapped [160].

Author had Elasticsearch 6.8.8 previously installed and now need to be upgraded to 7.6.2. According to Elasticsearch documentation, upgrading from 6.8.8 directly to 7.6.2 is possible. [161]

Documentation was not so good as Open Distro Security has. It was not so understandable and topics were not so easy to find. The author had problems with setting up Kibana.

After starting Kibana, it will optimize and cache browser bundles [162]. It may use all the memory from the system and might crash. Kibana log will let know by telling: "`FATAL ERROR: Ineffective mark compacts near heap limit Allocation failed - JavaScript heap out of memory`". In that case NODE_OPTIONS must be used in the end of the /etc/default/kibana file and set `NODE_OPTIONS="--max-old-space-size=2048"`.

Search Guard Elasticsearch can be upgraded if Elasticsearch is stopped and then old version of Search Guard removed [163]. New version of Search Guard can be installed as described in Appendix 3.

Search Guard Kibana plugin upgrade can be done by stopping Kibana, deleteing the Search Guard Kibana plugin from the plugins directory, restarting Kibana to clear the cached files, stopping Kibana and installing new version of the plugin.

Installing Search Guard took about 2 days, the steps have been described in Appendix 3. The author had issues with reading its documentation and problems with Kibana's configuration. Kibana optimization and cache browser bundles did not finished first, but the issue solved, if NODE_OPTIONS was limited in Kibana configuration.

Visualizations, dashboards, queries, searches can be exported or imported under Saved objects in Kibana Settings. Also canvas and map can be imported.

1 min load average per core was 0.725. CPU load results have been outlined more specifically in Figure 3 in the chapter 6. Elk-sgrd.test.uni uses 4 CPUs.

Adding new users to Search Guard is similar to Open Distro Security. Creating new role, adding it to action groups, choosing tenants and adding user to role mapping. In Search Guard if there is not something in the dropdown list, then it does not enable to add it directly. For example, if user is not predefined, admin has to create a new internal user and then go to role mapping and assign a previously created role to new user. In Open Distro, new users could be added in role mapping also.

Search Guard has additional feature called watches. It enables to send alerts if some conditions changes. Alerts can be sent to email, Slack, Jira, Pagerduty or webhook can be configured.

Moreover, Search Guard has Kibana space and feature controls. For example, SGS_XP_MACHINE_LEARNING and SGS_KIBANA_USER role can be assigned to user for using machine learning function in Kibana.

Search Guard audit logging has to be configured in elasticsearch.yml. By default, it is disabled [164]. The procedure is described in Appendix 3. The author is using default parameters of audit logging, but Search Guard enables to define audit log more specifically [164]. Audit logs are stored in `sg-7-auditlog*` indexes in Elasticsearch. To see audit logs in Kibana, index pattern has to be defined. It can be done from Management settings. Audit log shows permissions, index parameters, remote address, keywords, connection, user information, cluster name, host name, host address, node name and much more.

Search Guard API has GET, PUT, DELETE handlers for users, roles, roles mapping and action groups [165]. For example, list of internal users can be seen from Kibana Dev Tools with API GET request like this: `GET /_searchguard/api/internalusers/.`

5.3.3 Elasticsearch 7.6 with basic license

Elasticsearch documentation is spread and some information is repeated in different pages. This makes it difficult to find the best part of the documentation that needs to be implemented.

Upgrading Elasticsearch with its internal extensions is convenient, because all the packages are compatible with the new version and user does not have to check if

separate plugin is compatible with the newest version, if external plugins have not been used. Although, some features can be deprecated with the new version and deprecation log should be checked first. Also, reviewing breaking changes of Elasticsearch to check if configuration need to be changed [166]. Before an upgrade, it is important to check if data backup exists. Elasticsearch snapshots can be used to back up all the data [54]. Upgrading can be done using Elasticsearch repository and using Debian commands. Elasticsearch installation is described in Appendix 4.

The installation and configuring HTTPS certificates took 2 days and was smooth. Most of the time went on searching for instructions how configuration should be done.

Elasticsearch has script, which helps to make basic roles and generate passwords. With `elasticsearch-setup-passwords`, it is possible to create user for elastic, apm_system, kibana, logstash_system, beats_system and remote_monitoring_user. The procedure has been described in Appendix 4. After creating users, it is possible to log in to Kibana with elastic user. Built-in users are stored in Elasticsearch `.security` index [167]. All passwords are hashed in Elasticsearch security index.

The author used query `GET /.security/_search` in Kibana Dev Tools to look into the security index and to search users and their passwords. By default, Elasticsearch uses bcrypt hashing algorithm to store user passwords [168].

Every password can be changed through Kibana. Adding new users to Kibana is easy. It only needs username, password and which role to assign that user. New roles can be created, if default ones are not suitable. Role defines cluster privileges and index privileges, what kind of privilege on which indices. Moreover, space privileges can be defined, which assigns role to specific Kibana features.

Load average per 1 minute per core is 0.7155 according to Zabbix. Elk.test.uni uses 4 CPUs. Results have been outlined to figure 3 in chapter 6.

Data Visualizer in Kibana's Machine learning section enables to import data from log file directly and it helps to understand data and analyse it. CSV, JSON and log files with common format with timestamp can be imported. It is a new feature of Kibana but comes with Platinum and Enterprise versions and is experimental function of Kibana.

Dashboards, visualizations, saved searches can be exported or imported under Kibana's Saved Objects, which includes configuration, dashboards, visualizations, searches, canvas, map import or export possibility.

What is more, the API keys are created by the Elasticsearch API key service, which is enabled if TLS on the HTTP interface is configured. Another way is to enable `xpack.security.authc.api_key.enabled` setting. It is useful for access to API without requiring basic authentication [169].

6 Study results

Every plugin did what it was supposed to do. The list is in the Table 4. The requirements that were marked as unknown, existed as functionality that came out during testing. Open Distro Security did not have Map and Canvas import or export functionality, although configurations, dashboards, visualizations, queries and saved searches can be imported or exported.

Open Distro Security installation took the least time and had very good documentation that was easy to read and the topics were logically aligned.

Generating new users and assigning roles is the easiest with Elasticsearch X-Pack. No additional role mappings or tenants like role configuration in Search Guard and Open Distro Security have.

Upgrading is the easiest when no additional plugins are used for Elasticsearch, but if used, then compatibility between plugin and Elasticsearch versions has to be checked first.

Every plugin had Kibana spaces and feature controls with user roles or action group possibilities.

Every security tool has hashed passwords and is secure to use. They have several authentication options, HTTP basic authentication with certificates was tested in all three security tools.

The graph on Figure 3 shows the CPU load of the security plugins with 1 min average per core from 15:00:00 to 15:10:00 on May 13, 2020. Data has collected from internal Zabbix server.

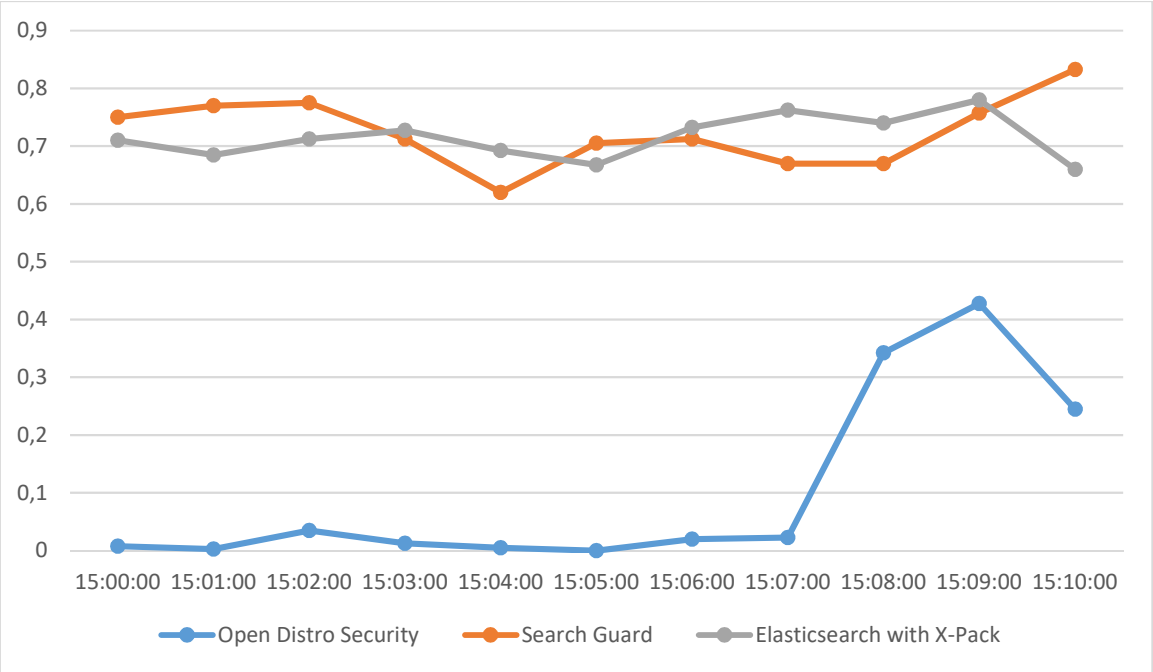

Figure 3. CPU load (1 min average per core).

The least CPU load had Elasticsearch with Open Distro Security. Open Distro Security maximum load was 0.4275, average load 0.0875 and minimum load 0.00 per core in 1 min. Search Guard for Elasticsearch had maximum load 0.8325, average load 0.725 and minimum load 0.62 per core in 1 min. Elasticsearch with X-Pack maximum value was 0.78, average load 0.7155 and minimum load 0.66 per core in 1 min.

Table 5 shows summarised results from chapter 5. The author will give 1 point in addition to security plugin if there is an advantage over other solutions per question. Advantages are marked with +1.

Table 5. Final results between Open Distro Security, Search Guard and Elasticsearch basic license with X-Pack.

Requirements and testing results	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Elastic-search basic license with X-Pack**
Time for installation	1 day (+1)	2 days	2 days
Installation difficulty	Easy (+1)	Installation had heap out of memory alert and installation did not finished at first	Easy (+1)
Plugin usage	New roles must have role mapping	New roles must have role mapping	Creating new users and roles is easiest (+1)
Kibana spaces and feature controls	+	+	+
Data import/export	+	+	+
Checking if it has all the features that it supposed to have from Table 4.	+	+	+

Requirements and testing results	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Elastic-search basic license with X-Pack**
Where passwords are stored?	System users in internal_users.yml and newly created users go to `.opendistro-_security` index	System users in sg_internal_users.yml, all users, roles and permissions in `_searchguard` index	Built-in users are stored in `.security` index.
Hashing algorithm	Default hashing algorithm not found; all passwords are hashed	Bcrypt hashing algorithm	Bcrypt hashing algorithm
CPU load	1 min average per core 0.0875 (+1)	1 min average per core 0.725	1 min average per core 0.7155
Documentation difficulty	Very good documentation (+1)	Not so good to read as Open Distro Security had	Different topics are spread
Differences between other plugins	API methods are: GET, PUT, POST, DELETE, PATCH	Additional feature called watches, which is for sending alerts. API methods are: GET, PUT, POST, DELETE, PATCH	API key service for access to API without requiring basic authentication

Requirements and testing results	**Open Distro Security for Elastic-search**	**Search Guard Enterprise version**	**Elastic-search basic license with X-Pack**
Difficulty of an upgrade	Plugin compatibility with Elasticsearch need to be checked first	Plugin compatibility with Elasticsearch need to be checked first	Easiest (+1)
Previous score + testing results = Full score	44 + 4 = 48	41 + 4 = 45 (additional 4 points for requirements that came out in testing and which were marked as unknown in Table 4)	32 + 3 = 35
The cost	No additional cost	2700 EUR per node annually	Basic version has no cost

The author recommends Open Distro Security for securing Elasticsearch and Kibana. Open Distro is an open source free tool that covers most of the requirements that were listed in Table 3 except customer support and is free to use for everyone. Open Distro Security has features that come with commercial solution, like LDAP, SAML, Kibana spaces and feature controls, field and document level security. Although, Open Distro is not yet available with Elasticsearch OSS 7.6.2, it is safe to use with Elasticsearch OSS 7.6.1, because Elasticsearch API keys are not used in Elasticsearch OSS. If cost is not an issue, then Search Guard Enterprise would be a good option that has customer support included also.

7 Recommendations for securing Elasticsearch

In this paragraph, the author states recommendations are given on how to securely store data in Elasticsearch and secure the access to the data in Kibana. Information is gathered together from the analysis and testing. These recommendations should be kept in mind when going to a production environment with Elasticsearch.

- Avoiding access from public networks to Elastic Stack. Elastic Stack should be behind firewall or VPN.
- Making regular snapshots of Elasticsearch data to back up its data.
- Using certificates to encrypt traffic between Elastic Stack components.
- Using authentication and authorization tool for sharing access to log information. Open Source Distro Security can be implemented without additional fees and has different authentication methods to choose the most suitable one for company.
- Setting up roles and groups with minimal user rights. User rights should only be granted based on the need for information to indexes and clusters.
- Protecting built in users with passwords. Built in users are elastic, kibana, logstash_system, beats_system, apm_system, remote_monitoring_user.
- Avoiding adding additional users to internal database. External user database should be preferred for better management of user permissions, for example using LDAP.
- Not using demo certificates in production environment.
- Monitoring Elastic Stack processes for getting to know that all components are alive and healthy.

- Auditing users for getting information what they do with Elasticsearch data and its components.
- Ensuring that access to configuration files and to private keys are not available for everyone who has access to system to protect its misuse.
- Not assigning 0.0.0.0 to access ports. This will let all clients to make connection to that port inside the network.

8 Future work

This thesis did not concentrate on hiding data values from logs. Another research question might be to find ways how to pseudonymize or anonymize values, so that system users or users with admin access would not see private data of service users in a centralized logging server. But, if the user requests to get data of his data-trace, then he would get it. This can be used in public services with many users.

9 Summary

The thesis started with a related work to this thesis and then with the description of methodology. The author used qualitative comparative analysis with the case study method to choose the best solutions for testing that would protect data in Elasticsearch and which use user groups in Kibana. Author used observational study to choose the best practical solution to secure data in the Elasticsearch. The analysis has been done by searching product documentations, different literature, comparing different solutions and by testing solutions in the authors' test environment to choose the best solution.

Then, the author gave a brief overview of logging and information that needs protection and searched what might happen if personal data loss is involved. For example, ICO issued a 204,6 million EUR fine to the British Airways according to GDPR, when its website diverted users' traffic to hacker website, which resulted in stealing personal data of more than 500 000 customers because of the poor security mechanisms.

Also, the author researched how common NoSQL databases have implemented security to protect their data. The author got to know that mostly NoSQL databases do not have enough security mechanisms with a default installation and security implementations needs to be configured to make database secured, for example in Elasticsearch, Apache CouchDB and MongoDB. The author found that OrientDB has strong security mechanisms and Apache CouchDB has the weakest security mechanisms. Different roles can be created in all of them. After reading up on NoSQL database security implementations and differences of Elasticsearch licenses, the author explored Elasticsearch components more specifically and searched for their use cases. For example, companies use Elasticsearch for their logs but also for websites like eBay for searching in their website and for social media platforms. Companies choose Elasticsearch, because it is fast in searching and is horizontally scalable solution that can be used to store main data, create searchable catalogue, document store and logging system.

Elastic Stack is Elasticsearch, Beats, Kibana and Logstash in all together. Elasticsearch stores information, Beats are transferring data to Elasticsearch, Logstash enables data transformation and Kibana has visualization tool for querying data, dashboards and graphs.

Author found that the most similar to Elastic Stack for logs is Graylog, but Graylog receives logs directly from application through the network protocol. Elastic Stack has tools to transfer logs from hosts to centralized logging server and it can read previously collected log files from the filesystem.

Then, the author searched for the biggest data breaches of Elasticsearch. Elasticsearch server have had data breaches with personal data loss. For example, on February 2020, French sports store had data leakage in which more than 123 million records leaked from improperly secured Elasticsearch server that contained private data. More examples can be found in the chapter 4.

After gathering enough background information of Elasticsearch, the author formed requirements for the security tools for Elasticsearch. The author compared Open Distro Security, Search Guard, ReadOnlyREST, Elasticsearch basic version with X-Pack features and Apache authentication in between them. The author chose out three open-source solutions for testing: Open-Distro Security, which is free to use for everyone; Search Guard enterprise version and Elasticsearch with X-Pack features.

The author has brought practical examples from testing on how to use security tools with Elasticsearch and chose the solution that would protect data in Elasticsearch. The best solution was chosen considered the requirements and results from the testing. The author has found that Open Distro Security for Elasticsearch fulfils the goal in a most effective way for companies who use Elasticsearch and Kibana in their centralized logging system. It enables role-based access control to Kibana with minimal user rights and secures access to Elasticsearch without additional fees and is free to use for everyone to protect data. In addition, Open Distro Security has features that come with commercial solutions.

The main contribution of this thesis were an analysis and tests for Elasticsearch security tools, recommendations are given on how to securely store data in Elasticsearch and secure the access to the data in Kibana, also configuration examples of security tools

that have been outlined in Appendixes. This study is different from other articles as it compared five most popular security solutions together for Elasticsearch by their features and evaluated three best solutions based on their functionalities practically. Also, insight into other NoSQL database security mechanisms was given.

References

[1] S. Arnus, "Providing Reliable Log Delivery and Integrity of Logs," TalTech Digikogu, 29.05.2017. [Online]. Available: https://digikogu.taltech.ee/et/Item/91293ef7-3c21-482a-8491-8d1c098dd288. [Accessed 1.11.2019].

[2] T. Hallas, "Logging Requirement Analysis and Specification for Development Based on governmental Institutions of Estonia," TalTech Digikogu, 23.01.2015. [Online]. Available: https://digikogu.taltech.ee/et/Item/856da267-18a9-40a1-84ea-95c6d447f32b. [Accessed 1.11.2019].

[3] S. Gupta and R. Rani, "Data Transformation and Query Analysis of Elasticsearch and CouchDB Document Oriented Databases," TIET Digital Repository, 31.08.2016. [Online]. Available: http://hdl.handle.net/10266/4215. [Accessed 03.12.2019].

[4] N. Prebensen, "A comparison of database systems for rapid search results in large amounts of leaked data," University of Agder, 2019. [Online]. Available: https://uia.brage.unit.no/uia-xmlui/handle/11250/2618708. [Accessed 02.11.2019].

[5] J. Moreno and E. B. Fernandez, "A Security Pattern for Key-Value NoSQL Database Authorization," in *EuroPLoP '18: Proceedings of the 23rd European Conference on Pattern Languages of Programs*, Irsee, 2018.

[6] W. Zugaj and A. S. Beichler, "Analysis of Standard Security Features for Selected NoSQL Systems," *American Journal of Information Science and Technology,* vol. 3, no. 2, pp. 41-49, 02.07.2019.

[7] P. M. Dhulavvagol, V. H. Bhajantri and S. G. Totad, "Performance Analysis of Distributed Processing System using Shard Selection Techniques on Elasticsearch," in *International Conference on Computational Intelligence and Data Science (ICCIDS 2019)*, 2020.

[8] S. J. Son and Y. Kwon, "Performance of ELK stack and commercial system in security log analysis," in *IEEE 13th Malaysia International Conference on Communications (MICC)*, Johor Bahru, 2017.

[9] W. Takase, T. Nakamura, Y. Watase and T. Sasaki, "A solution for secure use of Kibana and Elasticsearch in multi-user environment," 30.06.2017. [Online]. Available: https://arxiv.org/abs/1706.10040. [Accessed 17.05.2020].

[10] P. Kleindienst, "Building a real-world logging infrastructure with Logstash, Elasticsearch and Kibana," Stuttgart Media University, 2016. [Online]. Available: https://hdms.bsz-bw.de/frontdoor/deliver/index/docId/5021/file/elk_paper_patrick_kleindienst.pdf. [Accessed 17.05.2020].

[11] S. McCombes, "How to write a research methodology," 25.02.2020. [Online]. Available: https://www.scribbr.com/dissertation/methodology. [Accessed

20.11.2019].

[12] T. W. Edgar and D. O. Manz, in *Research Methods for Cyber Security*, Cambridge, United States, Todd Green, 2017, pp. 133-135.

[13] T. W. Edgar and D. Manz, in *Research Methods for Cyber Security*, Cambridge, United States, Todd Green, 2017, p. 90.

[14] "Information Literacy History: Search methods," [Online]. Available: https://libguides.rug.nl/c.php?g=470628&p=3218096. [Accessed 27.11.2019].

[15] "The benefits of the ELK stack without the operational overhead," Amazon Web Services, Inc., [Online]. Available: https://aws.amazon.com/elasticsearch-service/resources/articles/the-benefits-of-the-elk-stack/. [Accessed 05.11.2019].

[16] I. Wigmore, "Sensitive Information," 2014. [Online]. Available: https://whatis.techtarget.com/definition/sensitive-information. [Accessed 15.11.2019].

[17] "Personal Data," [Online]. Available: https://gdpr-info.eu/issues/personal-data/. [Accessed 16.11.2019].

[18] "What does the General Data Protection Regulation (GDPR) govern?," European Commission, [Online]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-does-general-data-protection-regulation-gdpr-govern_en. [Accessed 22.03.2020].

[19] "Protection of personal data," European Commission, [Online]. Available: https://ec.europa.eu/info/aid-development-cooperation-fundamental-rights/your-rights-eu/know-your-rights/freedoms/protection-personal-data_en. [Accessed 22.03.2020].

[20] "DLA Piper GDPR Data Breach Survey 2020," 20.01.2020. [Online]. Available: https://www.dlapiper.com/en/us/insights/publications/2020/01/gdpr-data-breach-survey-2020/. [Accessed 22.03.2020].

[21] "5 biggest GDPR fines so far [2020]," Data Privacy Manager, 04.02.2020. [Online]. Available: https://dataprivacymanager.net/5-biggest-gdpr-fines-so-far-2020/. [Accessed 22.03.2020].

[22] L. Issac, "SQL vs NoSQL Database Differences Explained with few Example DB," 14.01.2014. [Online]. Available: https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/. [Accessed 16.01.2020].

[23] "Top 12 NoSQL Document Databases," Pat Research, [Online]. Available: https://www.predictiveanalyticstoday.com/top-nosql-document-databases/. [Accessed 16.01.2020].

[24] M. Rouse, "What is authentication?," SearchSecurity, [Online]. Available: https://searchsecurity.techtarget.com/definition/authentication. [Accessed 19.01.2020].

[25] M. Rouse, "What is authorization?," SearchSoftwareQuality, [Online]. Available: https://searchsoftwarequality.techtarget.com/definition/authorization. [Accessed 19.01.2020].

[26] "Glossary," MongoDB manual, [Online]. Available: https://docs.mongodb.com/manual/reference/glossary/#term-least-privilege. [Accessed 20.01.2020].

[27] "Authentication," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/authentication/. [Accessed 20.01.2020].

[28] "Users," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/security-users/#user-authentication-database. [Accessed 22.01.2020].

[29] "Add Users," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/tutorial/create-users/. [Accessed 20.01.2020].

[30] "Authentication Mechanisms," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/authentication-mechanisms/#authentication-mechanism-default. [Accessed 20.01.2020].

[31] "SCRAM," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/security-scram/#authentication-scram. [Accessed 22.01.2020].

[32] "x.509," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/security-x.509. [Accessed 20.01.2020].

[33] "Configure mongod and mongos for TLS/SSL - MongoDB Manual," [Online]. Available: https://docs.mongodb.com/manual/tutorial/configure-ssl/. [Accessed 20.01.2020].

[34] "Enterprise Authentication Mechanisms," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/authentication-mechanisms-enterprise/#security-auth-kerberos. [Accessed 22.01.2020].

[35] "Distinguished Names and Relative Distinguished Names," 2010. [Online]. Available: https://docs.oracle.com/cd/E19182-01/820-6573/ghusi/index.html. [Accessed 03.03.2020].

[36] "Kerberos Authentication," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/kerberos/. [Accessed 25.01.2020].

[37] "Internal/Membership Authentication," MongoDB Manual, [Online]. Available: https://docs.mongodb.com/manual/core/security-internal-authentication/#inter-process-auth. [Accessed 22.01.2020].

[38] "SSL," OrientDB Manual, [Online]. Available: https://orientdb.com/docs/last/Using-SSL-with-OrientDB.html. [Accessed 16.01.2020].

[39] "Database Security," OrientDB Manual, [Online]. Available: https://orientdb.org/docs/3.1.x/security/Database-Security.html. [Accessed 18.01.2020].

[40] L. Gupta, "Java Secure Hashing – MD5, SHA256, SHA512, PBKDF2, BCrypt, SCrypt," HowToDoInJava, [Online]. Available: https://howtodoinjava.com/security/how-to-generate-secure-password-hash-md5-sha-pbkdf2-bcrypt-examples/. [Accessed 18.01.2020].

[41] "Database Encryption," OrientDB Manual, [Online]. Available: https://orientdb.com/docs/last/Database-Encryption.html. [Accessed 16.01.2020].

[42] "SSL," OrientDB Manual, [Online]. Available: https://orientdb.com/docs/last/Using-SSL-with-OrientDB.html. [Accessed 18.01.2020].

[43] "Apache CouchDB Documentation," [Online]. Available: https://docs.couchdb.org/en/stable/intro/security.html. [Accessed 18.01.2020].

[44] "Validation Funtions," CouchDB The Definitive Guide, [Online]. Available: http://guide.couchdb.org/draft/validation.html. [Accessed 18.01.2020].

[45] “At death’s door for years, widely used SHA1 function is now dead,” Arstechnica, 23.02.2017. [Online]. Available: https://arstechnica.com/information-technology/2017/02/at-deaths-door-for-years-widely-used-sha1-function-is-now-dead/. [Accessed 18.01.2020].

[46] "Adding Salt to Hashing: A Better Way to Store Passwords," 03.05.2018. [Online]. Available: https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/. [Accessed 18.01.2020].

[47] "Password Storage Cheat Sheet," OWASP, [Online]. Available: https://owasp.org/www-project-cheat-sheets/cheatsheets/Password_Storage_Cheat_Sheet.html#salting. [Accessed 18.01.2020].

[48] "Enterprise Security for Elasticsearch," [Online]. Available: https://www.elastic.co/what-is/elastic-stack-security?ultron=[EL]-[B]-[EMEA-General]-BMM&blade=adwords-s&Device=c&thor=%2Belasticsearch%20%2Bsecurity&gclid=Cj0KCQiAvJXxBRCeARIsAMSkAppJ1kSVSyP02JxDmGPezoW-G12Bem69OprRJpTSUDhRf9_XXspejVEaAg3ZEALw_wcB. [Accessed 25.01.2020].

[49] "What is the difference in Basic and open source license?," [Online]. Available: https://discuss.elastic.co/t/what-is-difference-in-basic-and-open-source-type-license/184313. [Accessed 26.01.2020].

[50] "7.1.0 release highlights," Elastic, [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.1/release-highlights-7.1.0.html. [Accessed 26.01.2020].

[51] "Subscriptions," [Online]. Available: https://www.elastic.co/subscriptions. [Accessed 26.01.2020].

[52] “REST APIs,” elastic, [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/rest-apis.html. [Accessed 04.04.2020].

[53] "Snapshot and restore," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshot-restore.html. [Accessed 28.01.2020].

[54] "Register a snapshot repository," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.5/snapshots-register-repository.html#snapshots-repository-plugins. [Accessed 28.01.2020].

[55] "Set up a cluster for high availability," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/high-availability.html. [Accessed 28.01.2020].

[56] "Search across clusters," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-cross-cluster-search.html. [Accessed 28.01.2020].

[57] "Node," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-node.html. [Accessed 28.01.2020].

[58] "Encrypting communications in Elasticsearch," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/configuring-tls.html. [Accessed 28.01.2020].

[59] "Configure role-based access control," [Online]. Available: https://www.elastic.co/guide/en/cloud-enterprise/current/ece-configure-rbac.html. [Accessed 28.01.2020].

[60] "File-based user authentication," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/file-realm.html. [Accessed 28.01.2020].

[61] "Native user authentication," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/native-realm.html. [Accessed 28.01.2020].

[62] A. Nadler, "Kibana Spaces: Organize (and Secure) Your Dashboards and Saved Objects," [Online]. Available: https://www.elastic.co/webinars/introduction-to-kibana-spaces. [Accessed 28.01.2020].

[63] L. Gregory, "Introducing Kibana feature controls: Curating and securing feature access," 26.06.2019. [Online]. Available: https://www.elastic.co/blog/introducing-kibana-feature-controls-curating-and-securing-feature-access. [Accessed 28.01.2020].

[64] "API Keys," [Online]. Available: https://www.elastic.co/guide/en/kibana/master/api-keys.html. [Accessed 28.01.2020].

[65] "Enabling audit logging," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.5/enable-audit-logging.html. [Accessed 28.01.2020].

[66] "Restricting connections with IP filtering," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/ip-filtering.html. [Accessed 28.01.2020].

[67] "Active Directory user authentication," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/master/active-directory-realm.html. [Accessed 28.01.2020].

[68] "PKI user authentication," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/pki-realm.html. [Accessed 28.01.2020].

[69] "Configuring SAML single-sign-on on the Elastic Stack," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.5/saml-guide.html. [Accessed 28.01.2020].

[70] "Document level security," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/document-level-security.html. [Accessed 28.01.2020].

[71] "Encryption at rest support in x-pack platinum," 11.09.2017. [Online]. Available: https://discuss.elastic.co/t/encryption-at-rest-support-in-x-pack-platinum/100111. [Accessed 28.01.2020].

[72] "Elastic Stack features," [Online]. Available: https://www.elastic.co/elastic-stack/features#full-stack-monitoring. [Accessed 28.01.2020].

[73] "Alerting," [Online]. Available: https://www.elastic.co/what-is/elasticsearch-alerting. [Accessed 28.01.2020].

[74] “Anomaly detection,” [Online]. Available: https://www.elastic.co/guide/en/kibana/current/xpack-ml-anomalies.html. [Accessed 28.01.2020].

[75] "Stories from users like you," elastic.co, [Online]. Available: https://www.elastic.co/customers/. [Accessed 09.10.2019].

[76] "Powering Uber Marketplace's Real-Time Data Needs with Elasticsearch," [Online]. Available: https://www.elastic.co/elasticon/conf/2017/sf/powering-uber-marketplace-real-time-data-needs-with-elasticsearch. [Accessed 07.05.2020].

[77] "eBay and Elasticsearch: This is not small data," [Online]. Available: https://www.elastic.co/videos/ebay-and-elasticsearch-this-is-not-small-data. [Accessed 07.05.2020].

[78] "From Hackathon to Production: Elasticsearch @ Facebook," [Online]. Available: https://www.elastic.co/elasticon/2015/sf/from-hackathon-to-production-elasticsearch-facebook. [Accessed 07.05.2020].

[79] G. P. Dritto, "An Overview on Elasticsearch and its usage," 27.03.2019. [Online]. Available: https://towardsdatascience.com/an-overview-on-elasticsearch-and-its-usage-e26df1d1d24a. [Accessed 07.05.2020].

[80] "Our Story," [Online]. Available: https://www.hgdata.com/about/. [Accessed 07.05.2020].

[81] "Find Out What Companies Are Using," [Online]. Available: https://discovery.hgdata.com/. [Accessed 07.05.2020].

[82] "Elastic Stack," [Online]. Available: https://www.elastic.co/what-is/elk-stack. [Accessed 05.11.2019].

[83] P. Sudheesh, "A Deep Dive Into Log Monitoring Using Elastic Stack," 20.01.2020. [Online]. Available: https://blog.qburst.com/2020/01/a-deep-dive-into-log-monitoring-using-elastic-stack/. [Accessed 27.04.2020].

[84] "Lightweight data shippers," [Online]. Available: https://www.elastic.co/products/beats. [Accessed 12.11.2019].

[85] G. P. Dritto, "An Overview on Elasticsearch and its usage," 27 03 2019. [Online]. Available: https://towardsdatascience.com/an-overview-on-elasticsearch-and-its-usage-e26df1d1d24a. [Accessed 09.10.2019].

[86] "Elasticsearch - Elastic Stack Tutorial (Part 2)," 15.01.2018. [Online]. Available: https://www.thecuriousdev.org/elasticsearch-elastic-stack-tutorial-part-2/. [Accessed 27.04.2020].

[87] Z. Tong, "What is an Elasticsearch Index?," 24.02.2013. [Online]. Available: https://www.elastic.co/blog/what-is-an-elasticsearch-index. [Accessed 12.11.2019].

[88] "Basic Consepts," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/6.2/_basic_concepts.html. [Accessed 12.11.2019].

[89] "Apache Solr," [Online]. Available: https://alternativeto.net/software/apache-solr/. [Accessed 14.11.2019].

[90] "What is the best alternative to ElasticSearch?," [Online]. Available: https://www.slant.co/options/358/alternatives/~elasticsearch-alternatives. [Accessed 14.11.2019].

[91] M. Peignon and M. Ganesh, "What are the top alternatives to Elasticsearch, and how do they compare?," 13.08.2017. [Online]. Available: https://www.quora.com/What-are-the-top-alternatives-to-Elasticsearch-and-how-do-they-compare. [Accessed 14.11.2019].

[92] "Log Management: Graylog vs ELK," JetRuby Agency, 17.01.2017. [Online]. Available: https://expertise.jetruby.com/log-management-graylog-vs-elk-d6e8f0492323. [Accessed 14.11.2019].

[93] "Advantages of Graylog+Grafana Compared to ELK Stack," 09.07.2018. [Online]. Available: https://medium.com/@logicify/advantages-of-graylog-grafana-compared-to-elk-stack-a7c86d58bc2c. [Accessed 29.04.2020].

[94] "ELK vs Graylog," stackshare, [Online]. Available: https://stackshare.io/stackups/elk-vs-graylog. [Accessed 15.11.2019].

[95] "ELK vs Logentries," stackshare, [Online]. Available: https://stackshare.io/stackups/elk-vs-logentries. [Accessed 15.11.2019].

[96] "Logentries," stackshare, [Online]. Available: https://stackshare.io/logentries. [Accessed 15.11.2019].

[97] "Logentreies," alternative.me, [Online]. Available: https://alternative.me/logentries#read_more. [Accessed 15.11.2019].

[98] "Pricing & Plans," rapid7, [Online]. Available: https://www.rapid7.com/products/insightops/pricing/. [Accessed 15.11.2019].

[99] "CISO Mag," CISOMAG, 20.03.2020. [Online]. Available: https://www.cisomag.com/unprotected-elasticsearch-server-leaks-5-billion-records/. [Accessed 21.03.2020].

[100] E. Targett, "Decathlon Leaks 123 Million Records via Insecure Elasticsearch Server," 25.02.2020. [Online]. Available: https://www.cbronline.com/news/decathlon-leaks. [Accessed 21.03.2020].

[101] V. Troia, "Personal And Social Information Of 1.2 Billion People Discovered In Massive Data Leak," 22.11.2019. [Online]. Available: https://www.dataviper.io/blog/2019/pdl-data-exposure-billion-people/. [Accessed 21.03.2020].

[102] P. Bischoff, "Report: 250 million Microsoft customer service and support records exposed on the web," Comparitech, 22.01.2020. [Online]. Available: https://www.comparitech.com/blog/information-security/microsoft-customer-service-data-leak/. [Accessed 21.03.2020].

[103] S. Gatlan, "Over 90 Million Records Leaked by Chinese Public Security Department," Bleepingcomputer, 08.07.2019. [Online]. Available: https://www.bleepingcomputer.com/news/security/over-90-million-records-leaked-by-chinese-public-security-department/. [Accessed 21.03.2020].

[104] C. Cimpanu, "ElasticSearch server exposed the personal data of over 57 million US citizens," ZDNet, 28.11.2018. [Online]. Available: https://www.zdnet.com/article/elasticsearch-server-exposed-the-personal-data-of-over-57-million-us-citizens/. [Accessed 21.03.2020].

[105] J. Bressers, "Tips to secure Elasticsearch clusters for free with encryption, users, and more," 04.06.2019. [Online]. Available: https://www.elastic.co/blog/tips-to-secure-elasticsearch-clusters-for-free-with-encryption-users-and-more. [Accessed 11.05.2020].

[106] "Open Distro for Elasticsearch," [Online]. Available: https://opendistro.github.io/for-elasticsearch/. [Accessed 09.02.2020].

[107] “Security,” Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/security.html. [Accessed 09.02.2020].

[108] "Alerting," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/alerting.html. [Accessed 09.02.2020].

[109] "Open Distro Security for Elasticsearch," [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/SQL%20Support.html. [Accessed 09.02.2020].

[110] “Performance Analyzer,” Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/analyzer.html. [Accessed 10.02.2020].

[111] "Open Distro for Security," [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/indexmanagement.html. [Accessed 10.02.2020].

[112] "New Features Coming Soon," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch/features/comingsoon.html. [Accessed 10.02.2020].

[113] "Security and Alertig for Elasticsearch and ELK stack," [Online]. Available: https://search-guard.com/. [Accessed 18.02.2020].

[114] "Licensing model," [Online]. Available: https://search-guard.com/licensing/#feature. [Accessed 18.02.2020].

[115] "Fewer decisions more security," [Online]. Available: https://readonlyrest.com/free/. [Accessed 11.05.2020].

[116] "Compare all the features," [Online]. Available: https://readonlyrest.com/#pricing. [Accessed 11.05.2020].

[117] "Key Features (ENTERPRISE)," [Online]. Available: https://readonlyrest.com/enterprise/. [Accessed 11.05.2020].

[118] “Apache Authentication and Authorization,” [Online]. Available: https://httpd.apache.org/docs/2.4/howto/auth.html. [Accessed 01.03.2020].

[119] "Apache Module mod_ssl," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_ssl.html. [Accessed 01.03.2020].

[120] "Apache Module mod_auth_digest," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_auth_digest.html. [Accessed 01.03.2020].

[121] "Apache Module mod_auth_basic," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_auth_basic.html. [Accessed 01.03.2020].

[122] "Apache Module mod_authn_dbm," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_authn_dbm.html. [Accessed 01.03.2020].

[123] "Apache Module mod_authn_file," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_authn_file.html. [Accessed 01.03.2020].

[124] "Apache Module mod_authn_dbd," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_authn_dbd.html. [Accessed 01.03.2020].

[125] "Apache Password Formats," [Online]. Available: https://httpd.apache.org/docs/2.4/misc/password_encryptions.html. [Accessed 01.03.2020].

[126] "Transport Layer Protection Cheat Sheet," [Online]. Available: https://owasp.org/www-project-cheat-sheets/cheatsheets/Transport_Layer_Protection_Cheat_Sheet. [Accessed 04.03.2020].

[127] "Linux Programmer's Manual," 30.04.2018. [Online]. Available: http://man7.org/linux/man-pages/man3/crypt.3.html. [Accessed 04.04.2020].

[128] "Apache Module mod_authnz_ldap," [Online]. Available: https://httpd.apache.org/docs/trunk/mod/mod_authnz_ldap.html. [Accessed 01.03.2020].

[129] "Apache Module mod_authn_socache," [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_authn_socache.html. [Accessed 01.03.2020].

[130] "SSL/TLS Strong Encryption: How-To," [Online]. Available: https://httpd.apache.org/docs/2.4/ssl/ssl_howto.html. [Accessed 04.03.2020].

[131] U. Vanem, "Configuring two-way SSL using Estonian EID smartcards in Ubuntu Apache2 webserver," 06.02.2019. [Online]. Available: https://www.id.ee/public/1912_-_Ubuntu_Apache2_webserver_SSL_configuration.pdf. [Accessed 04.03.2020].

[132] "Apache HTTP Server Support," [Online]. Available: https://httpd.apache.org/support.html. [Accessed 05.04.2020].

[133] "CVE-2019-13423," [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-13423. [Accessed 07.04.2020].

[134] "CVE-2020-7009," [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-7009. [Accessed 07.04.2020].

[135] "CVE-2020-1927," [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1927. [Accessed 07.04.2020].

[136] "CVE-2020-1934," [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1934. [Accessed 07.04.2020].

[137] "Open Distro for Elasticsearch Security plugin," [Online]. Available: https://github.com/opendistro-for-elasticsearch/security. [Accessed 02.05.2020].

[138] "Explore Gitlab," [Online]. Available: https://git.floragunn.com/public/. [Accessed 02.05.2020].

[139] "elasticsearch-readonlyrest-plugin," [Online]. Available: https://github.com/sscarduzio/elasticsearch-readonlyrest-plugin. [Accessed 11.05.2020].

[140] "Open Source, Distributed, RESTful Search Engine," [Online]. Available: https://github.com/elastic/elasticsearch. [Accessed 02.05.2020].

[141] "Mirror of Apache HTTP Server," [Online]. Available: https://github.com/apache/httpd. [Accessed 05.02.2020].

[142] "Open Distro for Elasticsearch Documentation," [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/. [Accessed 02.05.2020].

[143] "Search Guard 7.x-40.0.0 Documentation," [Online]. Available: https://docs.search-guard.com/latest/. [Accessed 02.05.2020].

[144] “Documentation for ReadonlyREST,” [Online]. Available: https://github.com/beshu-tech/readonlyrest-docs#documentation-for-readonlyrest-plugin. [Accessed 11.05.2020].

[145] "Elastic Stack and Product Document," [Online]. Available: https://www.elastic.co/guide/index.html. [Accessed 02.05.2020].

[146] "Apache HTTP Server Version 2.4 Documentation," [Online]. Available: http://httpd.apache.org/docs/current/. [Accessed 02.05.2020].

[147] "User Support and Discussion," [Online]. Available: https://httpd.apache.org/lists.html#http-users. [Accessed 02.05.2020].

[148] "Open Distro for Elasticsearch," [Online]. Available: https://discuss.opendistrocommunity.dev/. [Accessed 02.05.2020].

[149] "Search Guard Forum," [Online]. Available: https://forum.search-guard.com/latest/. [Accessed 02.05.2020].

[150] "Discuss the Elastic Stack," [Online]. Available: https://discuss.elastic.co/. [Accessed 02.05.2020].

[151] "Elastic product end of life dates," [Online]. Available: https://www.elastic.co/support/eol. [Accessed 06.04.2020].

[152] "Releases," [Online]. Available: https://wiki.ubuntu.com/Releases. [Accessed 07.05.2020].

[153] "Version history," [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/version-history/. [Accessed 21.04.2020].

[154] "Users and roles," [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/security-access-control/users-roles/. [Accessed 28.04.2020].

[155] "Backend configuration," [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/security-configuration/configuration/. [Accessed 28.04.2020].

[156] "API," [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/security-access-control/api/. [Accessed 29.04.2020].

[157] "Upgrade to 1.x.x," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/upgrade/1-0-0/. [Accessed 26.04.2020].

[158] "Configuring the Internal Users Database," [Online]. Available: https://docs.search-guard.com/latest/internal-users-database. [Accessed 15.04.2020].

[159] "Configuring authentication and authorization," [Online]. Available: https://docs.search-guard.com/latest/authentication-authorization. [Accessed 29.04.2020].

[160] "Active Directory and LDAP," [Online]. Available: https://docs.search-guard.com/latest/active-directory-ldap. [Accessed 15.04.2020].

[161] "Upgrade Elasticsearch," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-upgrade.html. [Accessed 22.04.2020].

[162] "Demo Installer," [Online]. Available: https://docs.search-guard.com/latest/demo-installer. [Accessed 14.04.2020].

[163] "Upgrading Search Guard," [Online]. Available: https://docs.search-guard.com/latest/upgrading. [Accessed 26.04.2020].

[164] "Audit Logging," [Online]. Available: https://docs.search-guard.com/latest/audit-logging-compliance. [Accessed 29.04.2020].

[165] "General usage and return values," [Online]. Available: https://docs.search-guard.com/latest/rest-api. [Accessed 29.04.2020].

[166] "Rolling upgrades," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/rolling-upgrades.html. [Accessed 26.04.2020].

[167] "Built-in users," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/built-in-users.html. [Accessed 28.04.2020].

[168] "Security setting in Elasticsearch," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/security-settings.html. [Accessed 28.04.2020].

[169] "Create API key API," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/security-api-create-api-key.html. [Accessed 29.04.2020].

[170] "Creating a Self-Signed SSL Certificate," Linuxize, 18.04.2019. [Online]. Available: https://linuxize.com/post/creating-a-self-signed-ssl-certificate/. [Accessed 13.04.2020].

[171] "Debian package," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/install/deb/. [Accessed 13.04.2020].

[172] "Kibana," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/kibana/. [Accessed 13.04.2020].

[173] "Encrypting communications in Kibana," [Online]. Available: https://www.elastic.co/guide/en/kibana/6.8/configuring-tls.html. [Accessed 15.04.2020].

[174] J. Handler, "Change your Admin Passwords in Open Distro for Elasticsearch," 21.03.2019. [Online]. Available: https://aws.amazon.com/blogs/opensource/change-passwords-open-distro-for-elasticsearch/. [Accessed 13.04.2020].

[175] "Apply configuration changes using securityadmin.sh," Open Distro for Elasticsearch, [Online]. Available: https://opendistro.github.io/for-elasticsearch-docs/docs/security-configuration/security-admin/. [Accessed 13.04.2020].

[176] "Search Guard Community Edition," [Online]. Available: https://docs.search-guard.com/latest/search-guard-community-edition. [Accessed 14.04.2020].

[177] "Using Kibana in a production environment," [Online]. Available: https://www.elastic.co/guide/en/kibana/7.6/production.html#memory. [Accessed 22.04.2020].

[178] "elasticsearch-setup-passwords," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/setup-passwords.html. [Accessed 18.04.2020].

[179] "Configuring security in Kibana," [Online]. Available: https://www.elastic.co/guide/en/kibana/current/using-kibana-with-security.html. [Accessed 18.04.2020].

[180] "Encrypting communications in Elasticsearch," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/configuring-tls.html. [Accessed 18.04.2020].

[181] R. Streefkerk, "Qualitative vs. quatitative research," 12.04.2019. [Online]. Available: https://www.scribbr.com/methodology/qualitative-quantitative-research/. [Accessed 20.11.2019].

[182] "Install Elasticsearch with Debian Package," [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/6.8/deb.html. [Accessed 13.04.2020].

Appendix 1 – Creating self-signed keys for HTTPS connection to Kibana

Author uses sign # if queries have been run by root user.

Generating self-signed certificates for web to use it later in Kibana for testing purposes

Installing openssl to generate a key and certificate:

apt install openssl [170]

Generating 4096-bit RSA key, x509 certificate with SHA 265 for 10 years:

openssl req -newkey rsa:4096 \

-x509 \

-sha256 \

-days 3650 \

-nodes \

-out kibana.crt \

-keyout kibana.key [170]

Printed output and answering the questions to get the certificate and a key:

Country Name (2 letter code) [AU]:EE

State or Province Name (full name) [Some-State]:Harjumaa

Locality Name (eg, city) []:Tallinn

Organization Name (eg, company) [Internet Widgits Pty Ltd]:TTÜ

Organizational Unit Name (eg, section) []:

Common Name (e.g. server FQDN or YOUR name) []:

Email Address []:mtamme@ttu.ee

Output: kibana.crt, kibana.key [170]

Moving certificate and a key to different directory and giving Kibana user permission to read the certificate and a key:

```
# mv kibana.key /etc/ssl/private/

# mv kibana.crt /etc/ssl/certs/

# chown kibana:kibana kibana.crt

# chown kibana:kibana kibana.key
```

Changing private key permissions, so that it would not be available for everyone to protect its misuse

```
# chmod 640 kibana.key
```

Recommendation for access permissions to configuration files. Access should be for the root user and for the service user, for example /etc/elasticsearch directory permissions.

```
# chown -R root:elasticsearch /etc/elasticsearch

# find /etc/elasticsearch -type d -exec chmod 750 {} \;

# find /etc/elasticsearch -type f -exec chmod 660 {} \;
```

Permissions for the elasticsearch certificates for internal usage:

```
# chown -R root:elasticsearch /etc/elasticsearch/certs

# chmod 600 /etc/elasticsearch/certs/CA_pw

# chmod 600 /etc/elasticsearch/certs/elastic-stack-ca.p12

# chmod 640 /etc/elasticsearch/certs/elastic-
certificates.p12
```

General recommendation for Elasticsearch: changing JVM parameters in elasticsearch configuration, so that it would have at least half of RAM.

```
# vim /etc/elasticsearch/jvm.options

-Xms2g

-Xmx2g
```

After changing parameters, elasticsearch service must be restarted.

Appendix 2 – Installing Open Distro for Elasticsearch

Author uses sign # if queries have been run by root user.

Installing Java 11:

```
# add-apt-repository ppa:openjdk-r/ppa
# apt update
# apt install openjdk-11-jdk
# apt install unzip
# export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
```

[171]

Adding repository key and downloading suitable repository

```
# wget -qO - https://d3g5vo6xdbdb9a.cloudfront.net/GPG-KEY-
opendistroforelasticsearch | sudo apt-key add -
# echo "deb https://d3g5vo6xdbdb9a.cloudfront.net/apt
stable main" | sudo tee -a
/etc/apt/sources.list.d/opendistroforelasticsearch.list
```

[171]

Installing elasticsearch-oss and Open Distro for Elasticsearch

Installing elasticsearch-oss 7.6.1:

```
# wget
https://artifacts.elastic.co/downloads/elasticsearch/elasti
csearch-oss-7.6.1-amd64.deb
# dpkg -i elasticsearch-oss-7.6.1-amd64.deb
```

[171]

Installing Open Distro for Elasticsearch and start the service:

```
# apt-get update
# apt install opendistroforelasticsearch
# systemctl start elasticsearch.service
```

[171]

Checking if Elasticsearch has been up and running:

```
# curl -XGET https://localhost:9200 -u admin:admin -
insecure
# curl -XGET https://localhost:9200/_cat/nodes?v -u
admin:admin –insecure
# curl -XGET https://localhost:9200/_cat/plugins?v -u
admin:admin –insecure
```

[171]

Adding Elasticsearch to startup folder:

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable elasticsearch.service
```

[171]

Open Distro Security for Kibana

Installing Open Distro for Kibana:

```
# apt install opendistroforelasticsearch-kibana
```

[172]

Modify Kibana configuration to use elasticsearch.hosts or elasticsearch.url:

```
# vim /etc/kibana/kibana.yml
```

[172]

Restart Kibana process and add it to stratup folder:

```
# systemctl start kibana.service

# systemctl daemon-reload

# systemctl enable kibana.service
```

[172]

Using previously generated openssl keys for web from Appendix 1 and changing Kibana configuration to use the keys:

```
# cat /etc/kibana/kibana.yml

server.host: 10.0.X.X

elasticsearch.hosts: https://localhost:9200

elasticsearch.ssl.verificationMode: none

elasticsearch.username: kibanaserver
```

```
# cat /etc/kibana/kibana.yml

elasticsearch.password: kibanaserver

elasticsearch.requestHeadersWhitelist:
["securitytenant","Authorization"]

opendistro_security.multitenancy.enabled: true

opendistro_security.multitenancy.tenants.preferred:
["Private", "Global"]

opendistro_security.readonly_mode.roles:
["kibana_read_only"]

##Additional configuration for SSL connection:

server.ssl.enabled: true

server.ssl.key: /etc/ssl/private/kibana.key

server.ssl.certificate: /etc/ssl/certs/kibana.crt

#elasticsearch.ssl.certificateAuthorities: /root/rootCA.pem
```

[173]

Restart Kibana service:

systemctl restart kibana.service

Kibana opens from port 5601:

https://10.0.x.x:5601

Default username: admin, password: admin

Changing internal user passwords of Open Distro Security

Changing default passwords:

```
# cd
/usr/share/elasticsearch/plugins/opendistro_security/tools

# bash hash.sh -p parool

$2y$12$C/543Qr4Y7Zy4Wsq5WvN9uw.WAbpvGghpiXvk9WexZgDfAGuG0OE
C
```

[174]

Copy hash to internal_users.yml:

```
# vim
/usr/share/elasticsearch/plugins/opendistro_security/securi
tyconfig/internal_users.yml [174]
```

Applying configuration changes:

```
# bash securityadmin.sh -cd ../securityconfig/ -icl -nhnv -
cacert /etc/elasticsearch/root-ca.pem -cert
/etc/elasticsearch/kirk.pem -key /etc/elasticsearch/kirk-
key.pem

[175]
```

Checking if password has been changed:

```
$ curl -XGET -k https://localhost:9200 -u admin:admin

Unauthorized

$ curl -XGET -k https://localhost:9200 -u admin:newpassword

{

  "name" : "elk-opd.test.uni",

  "cluster_name" : "elasticsearch",

  "cluster_uuid" : "isaCyPzNR6yy-n8epKSc8w",

  "version" : {

    "number" : "7.6.1",

    "build_flavor" : "oss",

    "build_type" : "deb",

    "build_hash" : "d2ef93dkbskdbksbdkabk23mskfbsk",

    "build_date" : "2020-02-29T00:15:25.529771Z",

    "build_snapshot" : false,

    "lucene_version" : "8.4.0",

    "minimum_wire_compatibility_version" : "6.8.0",

    "minimum_index_compatibility_version" : "6.0.0-beta1"

  },

  "tagline" : "You Know, for Search"

} [174]
```

New password needs to be given to kibana.yml file and restart the service:

```
# vim /etc/kibana/kibana.yml

elasticsearch.password: newpassword

# systemctl restart kibana
```

Appendix 3 – Installing Search Guard for ELK Stack

Author uses sign # if queries have been run by root user.

Suitable version of Search Guard for Elasticsearch, can be found from *https://docs.search-guard.com/latest/search-guard-versions*

Author uses Elasticsearch 7.2 for the Search Guard. This means that search-guard-7-7.6.2-40.0.0.zip will be used.

Search Guard installation

Installing Search Guard:

```
# cd /srv/
# wget https://releases.floragunn.com/search-guard-7/7.6.2-40.0.0/search-guard-7-7.6.2-40.0.0.zip
# wget https://releases.floragunn.com/search-guard-7/7.6.2-40.0.0/search-guard-7-7.6.2-40.0.0-sgadmin-standalone.zip
# wget https://releases.floragunn.com/search-guard-kibana-plugin-7/7.6.2-40.1.0/search-guard-kibana-plugin-7-7.6.2-40.1.0.zip
root@elk-sgrd:~# cd /usr/share/elasticsearch/
root@elk-sgrd:/usr/share/elasticsearch# bin/elasticsearch-plugin install -b file:///srv/search-guard-7/7.6.2-40.0.0/search-guard-7-7.6.2-40.0.0.zip
root@elk-sgrd:/usr/share/elasticsearch# cd plugins/search-guard-7/tools
root@elk-sgrd:/usr/share/elasticsearch/plugins/search-guard-6/tools# ./install_demo_configuration.sh
Search Guard 7 Demo Installer
#Only for testing purposes:
 ** Warning: Do not use on production or publicly reachable systems **
Install demo certificates? [y/N] y
Initialize Search Guard? [y/N] y
Enable cluster mode? [y/N] n
[162]
```

Checking if search guard is working: *https://localhost:9200/_searchguard/authinfo* [162].

Default username and password are admin:admin [162].

Changing elasticsearch.yml file with host IP to reach Elasticsearch remotely:

```
network.host: 10.0.x.x
```

As soon network host will be changed, elasticsearch starts to be in the production mode and wants to have discovery.seed_hosts, discovery.seed_providers, cluster.initial_master_node to be configured.

If free version of Search Guard is used, then elasticsearch.yml file should be changed to:

```
searchguard.enterprise_modules_enabled: false
```

[176]

After changing elasticsearch configuration, it must be restarted to set the changes:

```
# systemctl restart elasticsearch
```

Running sgadmin_demo.sh to read the contents of the configuration files in /usr/share/elasticsearch/plugins/search-guard-7/sgconfig and upload the contents to the Search Guard index:

```
# /usr/share/elasticsearch/plugins/search-guard-
7/tools/sgadmin_demo.sh
```

[162]

Configuring Audit logging

To enable audit logging the endpoint where audit events are stored has to be marked in elasticsearch.yml:

```
# vim /etc/elasticsearch/elasticsearch.yml
```

```
searchguard.audit.type: internal_elasticsearch
```

[164]

Search Guard Kibana

Suitable version of Search Guard for Kibana can be found from *https://docs.search-guard.com/latest/search-guard-versions.*

Author uses Kibana 7.6.2 and needs Search Guard Kibana 40.1.0 version for Kibana and downloads the package.

Installing Kibana:

```
# systemctl stop kibana
# cd /usr/share/kibana/
# /usr/share/kibana# bin/kibana-plugin install
file:///srv/search-guard-kibana-plugin-7-7.6.2-40.1.0.zip
```

Changing Kibana configuration to reach it remotely:

```
# Use HTTPS instead of HTTP
elasticsearch.hosts: "https://10.0.X.X:9200"

# Configure the Kibana internal server user
elasticsearch.username: "kibanaserver"
elasticsearch.password: "kibanaserver"

# Disable SSL verification when using self-signed demo
certificates
elasticsearch.ssl.verificationMode: none

# Whitelist the Search Guard Multi Tenancy Header
elasticsearch.requestHeadersWhitelist: [ "Authorization",
"sgtenant" ]

# X-Pack Security must be disabled, if Search Guard is
used:

xpack.security.enabled: false
[162]
```

Restarting Kibana service:

```
# systemctl restart kibana
```

After starting Kibana, it will optimize and cache browser bundles [162]. It may use all the memory from the system and might crash. In that case NODE_OPTIONS must be used in the end of the /etc/default/kibana file:

NODE_OPTIONS="--max-old-space-size=2048"

[177]

Kibana opens from *http://10.0.x.x:5601/* [162].

Default username and password are admin:admin [162].

Setting directory permissions for Kibana:

```
# chown -R kibana:kibana /usr/share/kibana/optimize/bundles
```

Changing default passwords

Internal users are configured in internal_users.yml file:

```
# less /usr/share/elasticsearch/plugins/search-guard-
6/sgconfig/sg_internal_users.yml
```
[158]

Generating new password hash:

```
# /usr/share/elasticsearch/plugins/search-guard-
6/tools/hash.sh -p newpassword
[158]
```

Adding previously generated hash to file sg_internal_users.yml and updating configuration:

```
root@elk-sgrd:/usr/share/elasticsearch/plugins/search-
guard-6/tools# ./sgadmin.sh -cd ../sgconfig/ -icl -nhnv -
cacert /etc/elasticsearch/root-ca.pem -cert
/etc/elasticsearch/kirk.pem -key /etc/elasticsearch/kirk-
key.pem
```

[158]

Allowing internal authentication:

```
basic_internal_auth_domain:
  http_enabled: true
  order: 1
  http_authenticator:
    type: basic
    challenge: true
  authentication_backend:
    type: internal
[158]
```

For authorization:

```
authz:
  internal_authorization:
    http_enabled: true
    authorization_backend:
      type: internal
[158]
```

Using previously generated certificates for Kibana and setting TLS/SSL connection

Adding previously generated certificates in Appendix 1 to kibana.yml file and enabling SSL connection:

```
server.ssl.enabled: true
server.ssl.key: /etc/ssl/private/kibana.key
server.ssl.certificate: /etc/ssl/certs/kibana.crt
#elasticsearch.ssl.certificateAuthorities: /root/rootCA.pem
[173]
```

After changing configuration, Elasticsearch and Kibana must be restarted.

Updating configuration:

```
root@elk-sgrd:/usr/share/elasticsearch/plugins/search-
guard-6/tools# ./sgadmin.sh -cd ../sgconfig/ -icl -nhnv -
cacert /etc/elasticsearch/root-ca.pem -cert
/etc/elasticsearch/kirk.pem -key /etc/elasticsearch/kirk-
key.pem
```

Appendix 4 – Installing Elasticsearch 7.6 with X-Pack using its' basic license

Author uses sign # if queries have been run by root user.

X-Pack

To configure authentication to elasticsearch, with HTTPS xpack security must be enabled in elasticsearch.yml file:

```
xpack.security.enabled: true
[178]
```

Creating basic roles

Creating elasticsearch basic roles and creating passwords, it can be with elasticsearch internal script:

```
#/usr/share/elasticsearch/bin/elasticsearch-setup-passwords
interactive
Initiating the setup of passwords for reserved users
elastic,apm_system,kibana,logstash_system,beats_system,remo
te_monitoring_user.
You will be prompted to enter passwords as the process
progresses.
Please confirm that you would like to continue [y/N]y


Enter password for [elastic]:
Reenter password for [elastic]:
Enter password for [apm_system]:
Reenter password for [apm_system]:
Enter password for [kibana]:
Reenter password for [kibana]:
Enter password for [logstash_system]:
Reenter password for [logstash_system]:
Enter password for [beats_system]:
Reenter password for [beats_system]:
Enter password for [remote_monitoring_user]:
Reenter password for [remote_monitoring_user]:
Changed password for user [apm_system]
[178]
```

```
Changed password for user [kibana]
Changed password for user [logstash_system]
Changed password for user [beats_system]
Changed password for user [remote_monitoring_user]
Changed password for user [elastic]
[178]
```

Testing elasticsearch connection:

```
# curl -u elastic --anyauth
'127.0.0.1:9200/_cat/nodes?pretty'
Enter host password for user 'elastic':
127.0.0.1 10 94 92 2.28 2.56 2.60 mdi * elk
```

Changing kibana.yml configuration, so that Kibana knows how to communicate with elasticsearch:

```
# vim /etc/kibana/kibana.yml

server.host: 10.0.X.X

elasticsearch.username: "kibana"
elasticsearch.password: "pass"
[179]
```

Restarting Kibana:

systemctl restart kibana

Trying to login to Kibana:

http://10.0.X.X:5601

Using elastic user role to login to Kibana with super admin rights.

Creating HTTPS connection between Kibana and Elasticsearch

Generating keys and making them available for elasticsearch user.

Creating CA [Certification Authority] certificate using elasticsearch-certutil:

```
# cd /usr/share/elasticsearch/
# bin/elasticsearch-certutil ca
[180]
```

Creating certificate and private key in PKCS#12 format:

```
# cd /usr/share/elasticsearch/
# bin/elasticsearch-certutil cert ca elastic-stack-ca.p12
[180]
```

Elastic-stack-ca.p12 contains public certificate for its' CA and private key for signing Elasticsearch nodes. [180]

Changing elasticsearch.yml to use certificates:

```
xpack.security.transport.ssl.enabled: true
xpack.security.transport.ssl.verification_mode: certificate
xpack.security.transport.ssl.keystore.path:
/etc/elasticsearch/certs/elastic-certificates.p12
xpack.security.transport.ssl.truststore.path:
/etc/elasticsearch/certs/elastic-certificates.p12
[180]
```

Changing permissions for the key, so that Elasticsearch can use the key:

```
# ls -la /etc/elasticsearch/certs/elastic-certificates.p12
-rw-r----- 1 root elasticsearch 3443 May 27 14:16
/etc/elasticsearch/certs/elastic-certificates.p12
```

Changing elasticsearch.yml, so that Kibana will have HTTPS connection with elasticsearch:

```
xpack.security.http.ssl.enabled: true
xpack.security.http.ssl.keystore.path:
/etc/elasticsearch/certs/elastic-certificates.p12
xpack.security.http.ssl.truststore.path:
/etc/elasticsearch/certs/elastic-certificates.p12
[180]
```

Restating elasticsearch:

```
# systemctl restart elasticsearch
```

Setting Kibana to use HTTPS certificates

Changing kibana.yml file to use previously generated certificates in Appendix 1 for web communication with end users

```
elasticsearch.hosts: ["https://127.0.0.1:9200/"]
xpack.security.enabled: true
server.ssl.enabled: true
server.ssl.key: /etc/ssl/private/kibana.key
server.ssl.certificate: /etc/ssl/certs/kibana.crt
#elasticsearch.ssl.certificateAuthorities:
#/etc/ssl/cacert/rootCA.pem

# For self-signed certificates, the verification mode shoud
be none if certificateAuthorities is not set:
elasticsearch.ssl.verificationMode: none
```

```
xpack.monitoring.elasticsearch.hosts:
["https://127.0.0.1:9200/"]
##xpack.monitoring.elasticsearch.ssl.certificateAuthorities
: /etc/ssl/cacert/rootCA.pem
[173]
```

Restarting kibana:

```
# systemctl restart kibana
```

Checking if HTTPS connection is working with elastic user:

```
# curl -k -u elastic --anyauth
'https://127.0.0.1:9200/_cat/nodes?pretty'
Enter host password for user 'elastic':
127.0.0.1 25 56 84 4.13 3.77 3.31 dim * elk


# curl -k -u kibana --anyauth 'https://127.0.0.1:9200/'
Enter host password for user 'kibana':
{
  "name" : "elk",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "Zf1JVwD2TfOHoEgMy5sKMQ",
  "version" : {
    "number" : "7.5.2",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "7a013degigibka438knsksl3",
    "build_date" : "2020-01-15T12:11:52.313576Z",
    "build_snapshot" : false,
    "lucene_version" : "8.3.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Accessing to Kibana using user elastic: https://10.0.X.X:5601/