Enterprise Grade Security and Data Protection for Big Data in the Cloud Using ZeroDB

Shayan Fazeli - Sepideh Berangi - Foad Jafari

Sharif University of Technology
shayan.fazeli@gmail.com

July 10, 2016

Abstract

The usage of big data analytics, a concept that nowadays everyone is familiar with, is growing continuously. Aside from all the advantages that this concept is offering, security is under debate. ZeroDB is a system like CryptDB that offers the benefits of database systems and the confidentiality of encryption schemas at the same time. As stated in zerodb's website: "ZeroDB enables enterprises to use the Cloud for storage and computation while keeping encryption keys on-premise.". In this approach we start working with this system from scratch.

I. Introduction

The confidentiality, authentication and integrity, are three main factors of a system's security. These factors must be addressed with encryption schemas that are secure to certain attacks. ZeroDB, uses these encryption schemas to provide us with a database system which has a layer of protection in front of it. In this project, we take a look at different aspects of this package.

II. Installation

In this part the installation of zerodb package is discussed in a step-by-step manner. Our version is installed on a specific linux distribution (ubuntu 16.04), but the installation procedure is pretty much the same for other linux distributions. First of all, python installer package known as "pip" must be collected and installed. This can be done using the following command:

```
sudo apt-get install python-pip
```

Note that we used python2.7 as our environment for installing zerodb. Now pip is installed and ready to be used, therefore, zerodb python package can be installed using this command:

```
sudo pip install zerodb
```

The version of zerodb which is available for pip will be detected automatically. When this is done, we need to clone the zerodb's repository from github. To do so, a new folder should be created. When in that new folder, one can insert this command to clone it:

```
git clone http://github.com/zerodb/
zerodb-server.git
```

After this phase, the requirements (like zerodb-server) must be installed. Change the

working directory using "cd" command to the "demo" folder inside zerodb's repository. In that, there is a file named "requirements.txt" that can be used as follows to install the necessary dependencies:

```
sudo pip install -r requirements.txt
```

Now, run this command:

```
zerodb-manager init_db
```

Now the user properties of the manager can be set. When this is done, zerodb is ready to use. Invoke zerodb using the following command:

```
zerodb-server
```

Now, using python, this database system can be manipulated.

III. Manipulation

A table can be defined as follows:

```
class Employee(Model):
    name = Field()
    surname = Field()
    description = Text()
    salary = Field()

    def __repr__(self):
        return
        "<%s %s who earns $%s>"
        % (self.name,
        self.surname,
        self.salary)
```

Using "create.py" script provided in demo folder, this table can be filled with many random values. Then, we head to the query part to see how we can ask for some information.

IV. Queries

A video of submitting queries is sent along with this report. Supported queries are listed below:

```
Contains(index_name, value)
Contains query.
Eq(index_name, value)
Equals query.
NotEq(index_name, value)
Not equal query
Gt(index_name, value)
Greater than query.
Lt(index_name, value)
Less than query.
Ge(index_name, value)
Greater (or equal) query.
Le(index_name, value)
Less (or equal) query.
DoesNotContain(index_name, value)
Does not contain query
Any(index_name, value)
Any of query.
NotAny(index_name, value)
Not any of query (ie, None of query)
All(index_name, value)
All query.
NotAll(index_name, value)
NotAll query.
InRange(index_name, start,
end, start_exclusive=False,
end_exclusive=False)
Index value falls within a range.
NotInRange(index_name, start,
end, start_exclusive=False,
end_exclusive=False)
Index value falls outside a range.
```

Logic operators are listed below as well:

```
$and: And
Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.
$or: Or
```

Joins query clauses with a logical OR returns all documents that match the conditions of either clause.
$not: Not
Inverts the effect of a query expression and returns documents that do not match the query expression.

And for field operations we have these operators:

$eq: Eq
Matches values that are equal to a specified value.
$ne: NotEq
Matches all values that are not equal to a specified value.
$lt: Lt
Matches values that are less than a specified value.
$lte: Le
Matches values that are less than or equal to a specified value.
$gt: Gt
Matches values that are greater than a specified value.
$gte: Ge
Matches values that are greater than or equal to a specified value.
$range: InRange
Matches values that fall within a specified range.
$nrange: NotInRange
Matches values that do not fall within a specified range.
$text: Contains
Performs text search for a specified value..
$ntext: DoesNotContain
Performs text search for the lack of a specified value.
$in: Any
Matches any of the values specified in an array.
$all: All
Matches arrays that contain all elements specified in the query.

$nany: NotAny
Matches none of the values specified in an array.
$nin: NotAll
Matches arrays that contain all elements specified in the query.

An script for testing simple queries is also written and is sent along.

References

[ZeroDB] http://zerodb.com