SECURITY ISSURES IN MANAGING UNSTRUCTURED DATA

Jim Q. Chen, Herberger Business School, St. Cloud State University, 720 Fourth Ave. South, St. Cloud, MN 56301, 320-308-4882, jchen@stcloudstate.edu
Alex Polacco, Herberger Business School, St. Cloud State University, 720 Fourth Ave. South, St. Cloud, MN 56301, 320-3008-3935, apolacco@stcloudstate.edu

ABSTRACT

This paper discusses security issues in managing unstructured big data. Potential security risks are identified in data collection, storage, and processing phase. Mitigation strategies will be suggested.

Keywords: Unstructured Data, NOSQL, Database Security, Big Data Security

INTRODUCTION

Unstructured data refers to data that exist in variety of formats such as sensor data, log files, data streams, documents, spreadsheets, text messages, emails, and facebook posts. They represent 80 percent of the exponential growth of global data, known as Big Data [1]. The world creates 2.5 Exabyte of data, equivalent of 2.5 billion Gigabytes of data per day [2]. The "big" in big data is commonly characterized by the 3 V's: Volume, Velocity, and Variety. Volume refers to the sheer size of data captured; velocity is the speed at which data are being accumulated and processed; variety is the diverse formats of data captured.

Traditional relational database management systems (RDBMS) are not suitable for managing unstructured data because of its lack of scalability and uniform data structure requirement. As a result, NoSQL databases have become a popular alternative technology to manage unstructured data.

However, with the rapid increase of NoSQL databases usage by business, securing those databases have become a vital issue. Recent data breaches occurred at MongoHQ and LinkedIn [3] [4] [5] underscore the importance of NoSQL data security. They point to a fact that NoSQL are becoming targets of attackers who seek valuable information.

While the concept of NoSQL (not only SQL) dates back to 1988, it has recently re-emerged in 2009 as a solution to managing big data [6]. NoSQL Databases can be divided into four groups [7]: Key-Values Databases, Column Databases, Document Oriented Databases, Graph Databases, XML databases. Those databases are designed without security as their first priority. Some offers very few security features.

This paper investigate security risks in the process of data collection, storage, processing, and transmission. Emphasis is given to security weaknesses of the most popular NoSQL databases. The paper suggests possible remedies. The assessment will be conducted along the following dimensions: authentication, authorization, data integrity, data file protection, client interface security, inter-cluster communication, NoSQL injection, and auditing.

DATA GENERATION, STORAGE, AND PROCESSING

Big data life cycle typically consists of six phases: plan, data capture, storage, processing, analysis and application [8]. In this paper, we focus on security issues in data capture, storage, and process phase.

Data capturing has increasingly become automated processes in business transactions and social interactions. With the rapid development of Internet of Things (IoT), huge amount of data are being collected via Radio Frequency Identification tags (RFID) and Wireless Sensor Networks (WSNs). The collected data are then transmitted on the Internet to its storage servers. Most of data are stored in NoSQL databases to be processed. Security breaches can occur in any of the above steps.

SECURITY ISSURES

Automated data capture can pose security risks due to the data collection devices' poor security design. Of particular important are the security risks in the use of RFID and WSNs for data collection in IoT. Large numbers of RFID tags are needed to implement Internet of Things applications. For economic reason, the tags are designed with low cost. They have small storage capacity and weak computational power [9]. Consequently, limited security features can be built in the tags.

WSNs are self-organizing networks consisting of spatially distributed autonomous devices using sensors to monitor physical or environmental conditions. Like RFID, WSNs have limited storage capacity, poor computing power, and narrow sensing range [9].

Unstructured data are stored and processed in NoSQL databases. According to database popularity ranking [10], the most widely used NoSQL database is MongoDB. However, Mongo DB servers have become most targeted databases recently. Thousands of MongoDB databases were compromised and held to ransom [3]. The following section will discuss MongoDb's system features and inherent security issues. MongoDB was first introduced in the early 2007 by MongoDB Inc. [11]. The name "MongoDB" was derived from the word "humongous", and reflects its original purpose: to create a database system that's able to support massive amount of data with relative ease [12]. By 2014, MongoDB had endorsements and supports from the four main cloud service providers: Google Cloud, Amazon AWS, Microsoft Azure, and Heroku [13].

MongoDB has a lot of performance and scalability-related features. However, it was not designed with security as its top requirement [11]. Its security weakness can be summarized as follows [14]:

1. Unencrypted Data Files: By default, MongoDB stores its information as unencrypted data files.
2. Insecure HTTP client interface: MongoDB uses a web-based interface that enables other applications to communicate with it. However, the interface is implemented as an insecure HTTP client interface.
3. SQL injection attack: Since it uses JavaScript as its "query language", it is susceptible to SQL injection attack.
4. Non-Universal Authentication: While MongoDB supports authentication in the standalone and replica-set mode, it does not support authentication in the sharded mode. In addition, MongoDB stores its password as an MD5 hash of the string <username>:mongo:<password>.
5. Non-universal authorization: When the authorization is defined, MongoDB only supports two user types, read-only and read-write. Each of these user types will have access to all data on the database collection. Any users that are defined within the admin database will have access to the entire database. By default, MongoDB does not provide a very granular, "need to know" authorization for its users.
6. Lack of audit: By default, MongoDB does not provide any auditing feature. It will log the creation of a new database collection, but not record any subsequent CRUD actions.

References available upon request from Jim Q. Chen.