Securing Databases Using Big Data Technology

JULIO MORENO, GSyA Research Group, Universidad de Castilla-La Mancha (UCLM)
MANUEL A. SERRANO, Alarcos Research Group, Universidad de Castilla-La Mancha (UCLM)
EDUARDO FERNÁNDEZ-MEDINA, GSyA Research Group, Universidad de Castilla-La Mancha (UCLM)

Abstract - Information is currently one of the most important assets for companies. This information is usually stored in databases, and preserving the security of these systems should, therefore, be a priority. Many different standards and methodologies explain how to achieve security objectives in an information system. We have carried out research based on the main standards and methodologies with the purpose of finding the principal controls and recommendations that can be applied in order to ensure the security of a database system. These controls were subsequently borne in mind during the design of our security evaluations. There was, however, still a problem: data are becoming more and more extensive, and we therefore needed a technology that would allow us to manage that amount of data efficiently so as to execute our evaluations. We thus decided to use the Big Data paradigm. The aim of this paper is to describe how Big Data technology can be used for evaluating the main security challenges of a database system, and the prototype we have implemented to accomplish that purpose.

• Technologies for IQ Improvement → Privacy & Security Issues.

Additional Key Words and Phrases: Security of data, Big Data, Relational Databases.

1. INTRODUCTION

Data is currently one of the most important assets for companies. It is essential as regards carrying out not only their daily activities, but also helping the businesses' managements to achieve their goals and make the best strategic decisions on the basis of the information extracted from that data (Kulkarni and Urolagin 2012). It is fundamental. Furthermore, storing important data in databases has long been a common practice of almost all companies (E. Bertino and R. Sandhu 2005).

Much of the data stored in databases is sensitive information that is attractive to data criminals, and there has consequently been a rise in the number of attacks that have occurred in order to access, modify or delete that data. Taking security measures in order to protect data has therefore become one of the biggest challenges for companies (Naghdi and Amini 2016). There are many different kinds of attacks that can affect the security of the database, and three main attributes must usually be achieved (Avižienis et al. 2004):

- Confidentiality: this is the absence of unauthorised disclosure information.
- Integrity: this is the absence of improper system alterations.
- Availability: this is the readiness for correct service.

Security is commonly known as a quality dimension that is present in the main quality models, but it is also a very important property that may have an impact on the idiosyncrasy of the data (Wang and Wang 2003). It is important for organisations to increase the quality and number of the controls used to minimise the effects resulting from attacks (Baker and Wallace 2007). This problem has been dealt with by many researchers and organisations that have created different standards and recommendations related to information security. Some of these standards can be applied to achieve a secure database system. In this paper, we base our research on the recommendations contained in the following standards: the ISO/IEC 27000-series concerning information security standards, ISO/IEC 15408, which deals with the

common criteria used to evaluate IT security, and COBIT 5 (Control Objectives for Information and related Technology).

Despite the existence of these standards and recommendations, there is still a problem when it is necessary to confront the challenge of evaluating any quality aspect in a database: the size of the database (Philip Chen and Zhang 2014). One of the main advantages of keeping data in a database is the quantity of data that can be stored there, but this may also be a great disadvantage as regards security. This occurs not only because it is more likely that criminals will become interested in attacking the database system, but also because the more data there are, the more the efficiency of a security evaluation algorithm decreases. This is the principal challenge in the world in which we live, since the data that we generate are increasing every day. For example, in 2012, we generated about 2.5 exabytes of data each day, and that number doubles every 40 months or so (McAfee and Brynjolfsson 2012). Furthermore, a few years ago we were concerned with creating a few hundred gigabytes and how to store them in our personal computers, while today it is necessary to think in terms of hundreds of terabytes (Kaisler et al. 2013). This tendency will not change in the near future, and it is estimated that the quantity of data that we create over a year will have to be measured with zettabytes (10^{18} bytes) (Harris 2008) (Yin and Kaynak 2015). More data implies that more time is needed in order to evaluate a quality aspect related to it. In order to manage all this data, we need a powerful analysis technique that can rapidly process a great amount of data. We therefore decided to solve our problem by using Big Data technology.

The term Big Data refers to a framework that allows the analysis and management of a larger amount of data than the traditional data processing technologies (Meng and Ci 2013). Big Data supposes a change from the traditional techniques in three different ways: the amount of data (volume), the rate of data generation and transmission (velocity) and the types of structured and unstructured data (variety) (Chen, Mao, and Liu 2014). These properties are known as the three basic V's of Big Data. Many authors have added new characteristics to the initial group, such as variability, veracity or value (Khan, Uddin, and Gupta 2014). This set of properties makes Big Data an appropriate technique with which to achieve the main goal of this paper.

The main goal of this paper is to describe a prototype with which to evaluate the security of a database system using Big Data techniques. The result of this process will be a final report that apprises of the security state of the system, according to the evaluations executed. This will be achieved by structuring the paper into different sections: we first present a brief introduction to the main challenges detected in the case of database security stated in the principal standards created for security, after which we describe the software architecture designed to achieve our goal, along with the evaluations already developed. Finally we present a section concerning our conclusions and future work.

2. MAIN CHALLENGES ON DATABASE SECURITY

Security is a very broad topic, with too many dimensions or subtopics to address as a whole. As explained in the previous section, the main goal that drives our work is that of implementing a prototype with which to evaluate the security of a database. In this paper we therefore decided to focus on the different standards and typical recommendations related to security that can be applied to the management of a

database. This section provides a brief review of the main controls and different recommendations carried out to achieve this.

2.1 ISO/IEC 15408

This standard (ISO and Std 2009), generally known as the Common Criteria standard, establishes the general concepts and principles of information technology evaluation. It also specifies a general evaluation model which is meant to be used as the basis for the evaluation of security properties. The Common Criteria ensure that the process of specification, implementation and evaluation of a computer security product has been carried out by means of a rigorous, standard, and repeatable process.

In order to certify a product with the Common Criteria framework, the product has to accomplish a large number of security parameters that have been accepted by 22 countries. The evaluation process checks the following aspects: that the product's requirements are correctly defined and implemented, and that the development process fulfils those requirements and is well documented.

2.2 ISO/IEC 27000-series

The ISO/IEC 27000-series (ISO and Std 2012) is a group of standards that have either been developed or are in progress created by ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission). The objective of creating this series of standards was to produce a framework with which to manage the information security in any organisation. The ISO/IEC 27000-series is formed of a large number of standards, and in our case we shall focus on those related to controls and recommendations that can be applied to a database system: ISO/IEC 27001 and ISO/IEC 27002.

ISO/IEC 27001 (ISO and Std 2013a) is a standard that provides the requirements and processes that an Information Security Management System (ISMS) must have. It emphasises the creation of a risk management approach in order to achieve effective information security. In summary, it provides the means to implement an effective information security management in compliance with the organisational objectives and business requirements (Humphreys 2006).

ISO/IEC 27002 (ISO and Std 2013b) is, meanwhile, a code of practice for information security management. ISO/IEC 27002 provides generic solutions that can be applied by any enterprise or organisation. This standard provides descriptions of a large number of security controls and their objectives, which are classified in 11 areas of information security management. Of these areas, we should like to highlight some that can be applied to improve the security in a database:

- Human resource security. This area describes controls with the objective of ensuring that employees and contractors understand their security responsibilities, are aware of the importance of information security, and are suitable for the roles that they are performing.
- Access control. The aim of this area is to limit access to information and information processing facilities. It also ensures that access is attained only by authorised users, thus preventing those who are not authorised from gaining access. The users are responsible for protecting their authentication information.

- Information security aspects of business continuity management. This area explains how information security continuity should be embedded in the organisation's business continuity management systems.

2.3 COBIT 5

COBIT 5 (ISACA 2012) is an information technology governance framework that provides a number of mechanisms with the objective of enabling the management to align business goals with IT goals. COBIT 5 achieves this by describing certain policies and good practises that can be used to control the technologies throughout the organisation.

In order to integrate the security into its model, COBIT 5 takes the BMIS (Business Model Information Security) as a basis and incorporates its comprehensive view and components into the new version. Thanks to this, COBIT 5 has a business oriented perspective for the management of information security.

COBIT 5 establishes a common language with which to refer to the protection of information. It changes the traditional vision of the need to make an investment in order to achieve information security. COBIT 5 is a risk management-based framework based on four different domains: Plan and Organise (PO), Acquire and Implement (AI), Deliver and Support (DS), and Monitor and Evaluate (ME). Each of these domains has its own controls. COBIT controls usually take into account the governance of business objectives, and it is for this reason that organisations tend to integrate it along with other standards such as ISO/IEC 27000 (Wolden, Valverde, and Talla 2015).

These guidelines indicate how the software should be implemented and how the security must be managed. These recommendations, along with the controls and best practises expressed in ISO/IEC 15408 and the ISO/IEC 27000 series, were very useful mechanisms as regards deciding which evaluations would be created in order to develop a prototype with which to evaluate the security of a database system.

3. ARCHITECTURE PROPOSED

In order to fulfil the requirements desired to accomplish our goal, we needed to develop an architecture that would allow us to create the different evaluations of which our tool would be formed. In this section, we describe the architecture eventually designed and how it works. This architecture is shown in Figure 1, which depicts the main components of our tool, including the database system, the Big Data environment, and the final report that is produced as a result of carrying out the evaluations.

The first component that we shall highlight is the Big Data environment. In our case, we decided to use Apache Hadoop. Hadoop is a framework developed by Apache that allows the distributed processing of large data sets across clusters of computers using programming models. It is designed to be scalable from a single server to thousands of them, each of which offers computation and local storage ('Apache Hadoop' 2016). Hadoop has its own distributed file system (HDFS) which stores the data in different servers with different functions, such as NameNode which is used to store the metadata or the DataNodes which store the application data (Shvachko et al. 2010). The principal characteristic of Hadoop is, however, that of being an open-source implementation of MapReduce (Jiang et al. 2010).

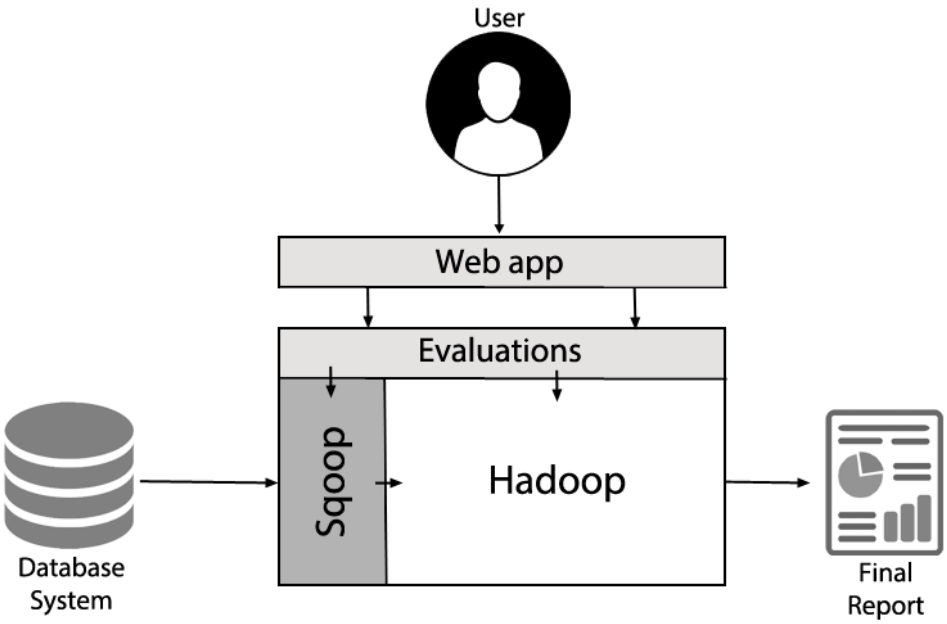

Fig. 1. Architecture design.

MapReduce is a programming model that is especially focused on processing and generating large data sets. The MapReduce paradigm accomplishes this by describing two different functions: the map function that processes the key/value pair needed to create a set of intermediate key/value pairs, and the reduce function that processes the intermediate values generated and merges them to produce a solution (Dean and Ghemawat 2004). This capacity to process large sets of data made the MapReduce model and the Hadoop implementation a good solution to our initial problem.

Another characteristic that justified our choice is the different applications that are available to increase the Hadoop functionalities. In our case, we needed an application with which to move the data stored in the database to the HDFS system in order to create the algorithms that our evaluations implement using the MapReduce programming model. This was done by using Sqoop. Apache Sqoop is a tool that allows the data to be bulked between Apache Hadoop and structured datastores such as a relational database ('Apache Sqoop' 2016).

Once we had obtained the structure required to store the data from the database in the Hadoop distributed file system, we needed a means to introduce the additional information required to perform the evaluations. For example, for some evaluations it is necessary to access maintenance tables, and we therefore need the information concerning the admin user. This problem was solved by implementing a web app based on the stack MEAN. Stack MEAN is a fullstack JavaScript framework which facilitates the creation of a web application ('Stack MEAN' 2016). This was done by employing the following technologies:

- MongoDB: a NoSQL database based on the concept of document rather than using the usual table format ('MongoDB' 2016). MongoDB is used to store the information required for the evaluations.

- Express: a minimal and flexible node.js web application framework ('Express' 2016) which provides a set of useful features for our app.
- AngularJS: a framework for web development focused on easily creating and maintaining web applications. It implements the Model-View-Whatever programming paradigm and was developed by Google ('AngularJS' 2016). AngularJS was used to create the front-end of our application.
- Node.js: a JavaScript runtime built on Chrome's V8 JavaScript engine which uses an asynchronous and event-driven model. One of its strength is its scalability. Node.js was used to develop the back-end of our application.

In order to present the results of the evaluations by means of a report, we decided to use the Google Charts extension that facilitates the visualisation of the data on a web page. It therefore includes different charts, such as a pie chart, a line chart, or a column chart ('Google Charts' 2016). These charts, along with some recommendations on how to improve the results of the security of the database, form the final report.

In summary, this architecture allows the user to introduce the data from the database into the Hadoop file system in order to execute the different evaluations available in the prototype. These evaluations will be implemented using the MapReduce programming model. The user needs to interact with the prototype using a web application in order to provide the information required by the evaluation chosen. A final report containing the results of the evaluations will then be generated by the system.

4. DEVELOPED EVALUATIONS

Once we had decided on our architecture, we then went on to create the different security evaluations of which our system is formed. The decision to choose these evaluations was made on the basis of the main controls and guidelines described in the Section 2 of this paper. In this section, we shall describe the different evaluations: what their motivation is, and how we implement them. In this first prototype of our tool, we have implemented five different evaluations, which are summarised in Table I.

Table I. Developed evaluations.

Id	Evaluation	Aim	Motivation
EV 1	To evaluate the encryption of a column in a database table.	To check that the fields of a column are encrypted.	ISO/IEC 27002 explains the need to have a policy on the use of encryption of the data.
EV 2	To evaluate the users' permissions.	To check which users have the proper permissions.	COBIT 5 proposes the application of the least privilege principle to ensure the security of a system.
EV 3	To evaluate that deleted users have no privileges.	To check whether users that are not part of the system have been properly deleted.	ISO/IEC 27002 urges organisations to protect their interests when users no longer belong to the system.

EV 4	To evaluate that a number of users only gain access during their work time.	To check that users only gain access during the time they are allowed to do so.	Inside attacks made by users that are part of the system is one the main sources of threats (Scopinaro 2012).
EV 5	To evaluate the quantity of failed accesses that have occurred.	To check how many wrong accesses have been made by each user.	ISO 15408 describes the need for users to have authentication and identification controls.

These evaluations were created using the MapReduce programming model, which allows the management of large amounts of data. For example, in order to carry out these evaluations it is necessary to analyse the tables of a database or log files, which may be huge. Figure 2 shows a diagram that explains the usual process followed to execute an evaluation in our prototype.

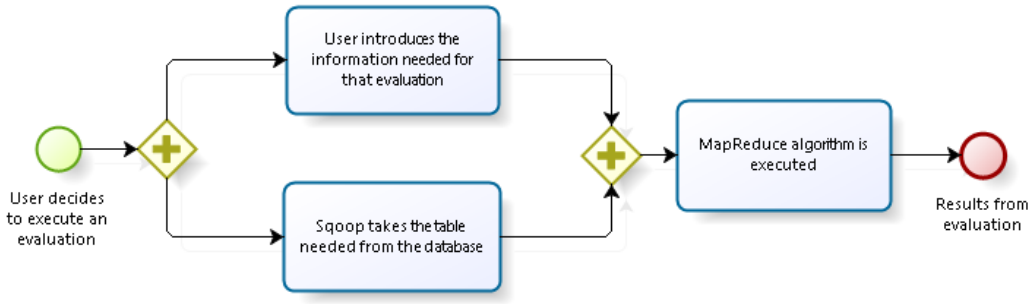

Fig. 2. Process diagram of evaluation 2.

4.1 Evaluation 1 – Encryption.

This evaluation checks whether a column provided by the user is encrypted. As a result of effectuating this evaluation, a percentage of the fields encrypted is obtained. If the column we are evaluating contains sensitive or relevant data, it is important to make some kind of encryption in order to protect it.

We decided to do this by making two different dictionaries available in our system: one in Spanish and another in English. Having included these dictionaries, we then created a MapReduce algorithm that compares each word of every field contained in the column with the dictionaries. In order to obtain the selected column in the HDFS we made use of the Apache Sqoop tool. If a word in the field is not recognised as a proper word, we consider that the field is encrypted. Otherwise, the field is not considered to be encrypted.

4.2 Evaluation 2 – Users permissions.

This evaluation was developed with the objective of evaluating the users' permission. It can be considered as an implementation of the principle of least privilege explained in the second section of this paper.

This objective was accomplished by implementing a MapReduce algorithm whose input is the table in which the users' permissions are stored. For example, in a MySQL database the USER_PRIVILEGES are located with the metadata from the database. Once the algorithm has been executed, a percentage of the users with a particular privilege will be shown as a result. A large quantity of users with a particular privilege may be a sign of vulnerability in the system.

4.3 Evaluation 3 – Deleted users.

The aim of implementing this evaluation was to attain a specific goal: to detect whether users that no longer belong to the system still have privileges in it. For example, this situation may occur when a worker is dismissed but his/her privileges are not removed from the system. Having former users with privileges exposes our system to attacks from people that are no longer part of the company, which can be a great threat.

In order to complete this task, we need a list of the users that should no longer be in the system. This list must be provided by the user. This list and the USER_PRIVILEGES table are used to create a MapReduce algorithm which creates an output with the percentage of users that still have any privileges in the system.

4.4 Evaluation 4 – Access in work time.

In general, users should only access the system when they are working. So, if a user repeatedly accesses the system when s/he is not working, this may indicate an inappropriate activity. The objective of this evaluation is to check how many times this has occurred and which users have done so.

In order to achieve that goal, we need two things as input for our MapReduce algorithm: a file with the users' schedules and the log file automatically generated by the database system. Once we have executed the evaluation, we will obtain a sorted list of those users that access the system when not working.

4.5 Evaluation 5 – Wrong accesses.

The main objective of this evaluation is to check the number of wrong accesses made by each user. The reason behind this evaluation is that if some users have made a lot of mistakes, this may be indicative of an attempt to attack the system. This type of attack may be perpetrated by the user him/herself or by an outsider trying to access the system.

This goal can be attained by simply using the log file generated by the database system. Once this log has been obtained, it is analysed using a MapReduce algorithm. The result of executing the evaluation will be the number of times that each user has failed in an attempt to access the system.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have described a prototype created with the objective of evaluating different security aspects in a database. We have therefore described the main challenges, techniques, and mechanisms that the principal standards recommend in order to attain a secure system. Bearing this in mind, we designed five different evaluations that make use of the Big Data technologies, specifically MapReduce algorithms running on a Hadoop environment. We made the decision to use Big Data owing to the tendency of data, and consequently of databases, to increase in size.

This prototype was conceived with the idea of it being easily extensible with new evaluations, not only in the field of security, but also for other data quality purposes. We are considering the implementation of some new evaluations, along with certain improvements that will allow us to create an interesting tool. Furthermore, we wish to deploy our prototype in a bigger cluster that would allow us to properly check whether our system improves the time it takes to execute an evaluation of a large amount of data.

ACKNOWLEDGEMENTS

This work has been funded by the SEQUOIA project (Ministerio de Economía y Competitividad and FondoEuropeo de Desarrollo Regional FEDER, TIN2015-63502-C3-1-R) and by the SERENIDAD project (Consejería de Educación, Ciencia y Cultura de la Junta de Comunidades de Castilla La Mancha, y Fondo Europeo de Desarrollo Regional FEDER, PEII-2014-045-P).

REFERENCES

'AngularJS'. 2016. Accessed March 14. https://angularjs.org/.
'Apache Hadoop'. 2016. Accessed March 14. http://hadoop.apache.org/.
'Apache Sqoop'. 2016. Accessed March 14. http://sqoop.apache.org/.
Avižienis, A., J.-C. Laprie, B. Randell, and C. Landwehr. 2004. 'Basic Concepts and Taxonomy of Dependable and Secure Computing'. *IEEE Transactions on Dependable and Secure Computing* 1 (1): 11–33. doi:10.1109/TDSC.2004.2.
Baker, W.H., and L. Wallace. 2007. 'Is Information Security under Control?: Investigating Quality in Information Security Management'. *IEEE Security and Privacy* 5 (1): 36–44. doi:10.1109/MSP.2007.11.
Chen, M., S. Mao, and Y. Liu. 2014. 'Big Data: A Survey'. *Mobile Networks and Applications* 19 (2): 171–209. doi:10.1007/s11036-013-0489-0.
Dean, Jeffrey, and Sanjay Ghemawat. 2004. 'MapReduce: Simplified Data Processing on Large Clusters'. *Commun. ACM* 51 (1): 107–13. doi:10.1145/1327452.1327492.
E. Bertino, and R. Sandhu. 2005. 'Database Security - Concepts, Approaches, and Challenges'. *IEEE Transactions on Dependable and Secure Computing* 2 (1): 2–19. doi:10.1109/TDSC.2005.9.
'Express'. 2016. Accessed March 14. http://expressjs.com/.
'Google Charts'. 2016. *Google Developers - Charts*. Accessed March 14. https://developers.google.com/chart/.
Harris, D. 2008. 'Storage Must Prepare for the Zettabyte Universe'. *Electronic Design* 56 (8): 45–47.
Humphreys, Ted. 2006. 'State-of-the-Art Information Security Management Systems with ISO/IEC 27001: 2005'. *ISO Management Systems* 6: 1.
ISACA, Information Systems Audit and Control Association. 2012. *Cobit 5 A Business Framework for the Governance and Management of Enterprise*. ISACA.
ISO, and I. E. C. Std. 2009. 'ISO/IEC 15408-1: 2009'. *Information Technology-Security Techniques-Evaluation Criteria for IT Security-Part* 1.
ISO, and I. E. C. Std. 2012. 'ISO/IEC 27000:2012-Series'. *Information Technology. Security Techniques. Information Security Management Systems. Overview and Vocabulary.* 1.
ISO, and I. E. C. Std. 2013a. 'ISO/IEC 27001:2013'. *Information Technology. Security Techniques. Information Security Management Systems. Requirements.* 1.
ISO, and I. E. C. Std. 2013b. 'ISO/IEC 27002:2013'. *Information Technology. Security Techniques. Code of Practise for Information Security Controls.* 1.
Jiang, D., B.C. Ooi, L. Shi, and S. Wu. 2010. 'The Performance of Mapreduce: An Indepth Study'. *Proceedings of the VLDB Endowment* 3 (1): 472–83.

Kaisler, S., F. Armour, J.A. Espinosa, and W. Money. 2013. 'Big Data: Issues and Challenges Moving Forward'. In , 995–1004. doi:10.1109/HICSS.2013.645.
Khan, M.A.-U.-D., M.F. Uddin, and N. Gupta. 2014. 'Seven V's of Big Data Understanding Big Data to Extract Value'. In . doi:10.1109/ASEEZone1.2014.6820689.
Kulkarni, Mr Saurabh, and Dr Siddhaling Urolagin. 2012. 'Review of Attacks on Databases and Database Security Techniques'. *International Journal of Emerging Technology and Advanced Engineering, ISSN*, 2250–2459.
McAfee, A., and E. Brynjolfsson. 2012. 'Big Data: The Management Revolution.' *Harvard Business Review* 90 (10): 60–66, 68, 128.
Meng, X., and X. Ci. 2013. 'Big Data Management: Concepts, Techniques and Challenges'. *Jisuanji Yanjiu Yu Fazhan / Computer Research and Development* 50 (1): 146–69.
'MongoDB'. 2016. *MongoDB*. Accessed March 14. https://www.mongodb.com/.
Naghdi, S., and M. Amini. 2016. 'Preventing Database Schema Extraction by Error Message Handling'. *Information Systems* 56: 135–56. doi:10.1016/j.is.2015.09.010.
Philip Chen, C.L., and C.-Y. Zhang. 2014. 'Data-Intensive Applications, Challenges, Techniques and Technologies: A Survey on Big Data'. *Information Sciences* 275: 314–47. doi:10.1016/j.ins.2014.01.015.
Scopinaro, N. 2012. 'Defending Our Database against inside Attacks'. *Surgery for Obesity and Related Diseases* 8 (6): 814. doi:10.1016/j.soard.2012.08.005.
Shvachko, K., H. Kuang, S. Radia, and R. Chansler. 2010. 'The Hadoop Distributed File System'. In . doi:10.1109/MSST.2010.5496972.
'Stack MEAN'. 2016. *MEAN.IO*. Accessed March 14. http://mean.io.
Wang, H., and C. Wang. 2003. 'Taxonomy of Security Considerations and Software Quality'. *Communications of the ACM* 46 (6): 75–78. doi:10.1145/777313.777315.
Wolden, M., R. Valverde, and M. Talla. 2015. 'The Effectiveness of COBIT 5 Information Security Framework for Reducing Cyber Attacks on Supply Chain Management System'. In , 48:1846–52. doi:10.1016/j.ifacol.2015.06.355.
Yin, S., and O. Kaynak. 2015. 'Big Data for Modern Industry: Challenges and Trends'. *Proceedings of the IEEE* 103 (2): 143–46. doi:10.1109/JPROC.2015.2388958.