International Journal of Information and Communication Sciences
2021; 6(3): 62-65
http://www.sciencepublishinggroup.com/j/ijics
doi: 10.11648/j.ijics.20210603.12
ISSN: 2575-1700 (Print); ISSN: 2575-1719 (Online)

Data Sensitivity and Classification Management: A Declarative Approach

Yuejin Zhang[1], Hong Liu[2, 3], Guowei Wang[4]

[1]Department of Electrical Engineering, Smart City College, Beijing Union University, Beijing, China

[2]Run Technologies Co., Ltd. Beijing, Beijing, China

[3]Beijing Cyberspace Data Analysis and Applied Engineering Technology Research Center, Beijing, China

[4]Beijing Municipal Bureau of Public Security, Beijing, China

Email address:
herryyj707@vip.sina.com (Yuejin Zhang)

To cite this article:
Yuejin Zhang, Hong Liu, Guowei Wang. Data Sensitivity and Classification Management: A Declarative Approach. *International Journal of Information and Communication Sciences*. Vol. 6, No. 3, 2021, pp. 62-65. doi: 10.11648/j.ijics.20210603.12

Received: July 12, 2021; **Accepted**: July 30, 2021; **Published**: August 9, 2021

Abstract: Data protection according to sensitivity and classification has become a mandatory security mechanism for safety- and security-critical organizations. There is however no consensus on how to implement data sensitivity and classification in existing big-data systems. An approach is proposed to express and compute data sensitivity and multidimensional data classification in fine granularity. The approach is based on a declarative logic programming language, which is able to separate security requirement definitions and deduction from implementation details. Expressing and validating the security rules can be done transparently, ignoring underlying technical migrations and infrastructure differences. It is therefore possible to use the same set of security rules among various big data systems. Compared to other logic-programming-based approach, the declarative nature also makes it preferable for modular development and system maintenance. Sensitivity specification is shown and security analysis including conflict detection and resolution is performed on the same platform. Several typical types of data classification have also been illustrated and analyzed. The approach is capable of expressing complex classification methods, including classification with multiple parameters, classification according to graph computation, and classification based on relations among multiple data objects. The logic programming-based method is shown to have more expressive power and better complexity performance than conventional methods.

Keywords: Data Security, Sensitivity and Classification, Logic Programming, Big Data

1. Introduction

Recently, the Data Security Law has been published and is to take effect in September. Data have become one of the most important assets in an organization, and data security is crucial for both the safety and the development of a society. The Data Security Law requires that data should be protected according to data sensitivity and classification, which is determined based on the harm of a potential data breach to national security, public interests, or legal rights of individuals or organizations.

Although the Data Security Law does not specify the systems or techniques to enforce data sensitivity and classification management, data sensitivity levels are in general used to realize conventional Mandatory Access Control (MAC)-like security mechanisms, and multi-dimensional data classification types are used as alternatives to control data access in a less constrained way.

Data sensitivity rules describe the rules to assign data to a sensitivity level. There are multiple ways to design sensitivity rules. The simplest is to assign sensitivity levels according to the attribute/column names. Only visitors with equal or higher sensitivity levels will be allowed to read the columns.

Data classification is to assign data types from multiple points of view. The same data can be assigned several different types for use in different applications. Classification may be set according to data content or statistic characteristics, by for example recognizing patterns in the data or related data. Data protection can also be enforced according to data types. In contrast with sensitivity, which is often a global value, data classification is more flexible and can support data protection

mechanisms other than MAC.

In practice, data sensitivity and classification configuration is often implemented using labels [1], because most big data applications already have a label system. Adding security features in this way may incur very little cost and changes to existing systems. However, label systems have very limited expressive power, and are in practice difficult for developers and users to understand, analyze, and use, as labels often lack formal definitions.

A declarative approach is proposed to implement data sensitivity and classification configuration. The approach is based on the logic programming language Datalog. A declarative language separates business logic from technical details, which is beneficial to today's ever-changing big data applications. The method utilizes Datalog to provide a formal framework for data sensitivity and classification specifications, which is concise and easy to understand.

2. Related Work

Although the Data Security Law has made clear requirements for sensitivity and classification-based data security and individual information protection, it does not suggest any specific technical methods on how enterprises deploy these data protection mechanisms or systems. Currently there are no standard development methods, processes, or evaluation tools for data sensitivity and classification. Big data corporations often improvise their own ways to realize sensitivity and classification-based data security, whose correctness and effectiveness is questionable. [2]

Here a logic-programming based approach is used to manage data sensitivity and classification. The main goal is to express and analyze multi-dimensional data sensitivity and classification, and to provide separation of business logic or security requirement from implementations. The language used is Datalog, which is a declarative subset of Prolog. In particular, the order of Datalog clauses is not relevant to its logical consequences. These years important progress has been made on developing high-performance in-memory and distributed parallel Datalog reasoners, as well as the improvement on the language expressiveness. Datalog has therefore regained notice in academia and in industry. Datalog-based systems such as LogicBlox [3], RDFox [4], BigDatalog [5], RaSQL [6], are developed and used to perform big data analysis, graph computing, knowledge reasoning, security, and business decision making. [7-9]

3. The Framework

The declarative approach to manage data sensitivity and classification is to insert a declarative formal form of definitions and rules of sensitivity and multi-dimensional classification between security requirements and underlying implementations. Security analysts are responsible for converting human understanding of sensitivity and classification-based data protection into Datalog rules, analyze and validate the configuration. Different applications use their own classification rules independently. Engine developers are responsible for maintaining the semantic equivalence of Datalog programs and the underlying big data systems. For example, data stored in the relational databases, graph databases, document databases, etc., must be converted into Datalog facts.

After the data sensitivity and classification rules are defined, data protection mechanisms can be further deployed based on the results of Datalog programs, or even be defined within the same framework. One way of using the declarative approach is to compute all the results of sensitivity and classification for the given database, which corresponds to materialization. On the other hand, if rules or data are changed frequently, query algorithms can be used to compute sensitivity and classification types for specific data objects on the fly.

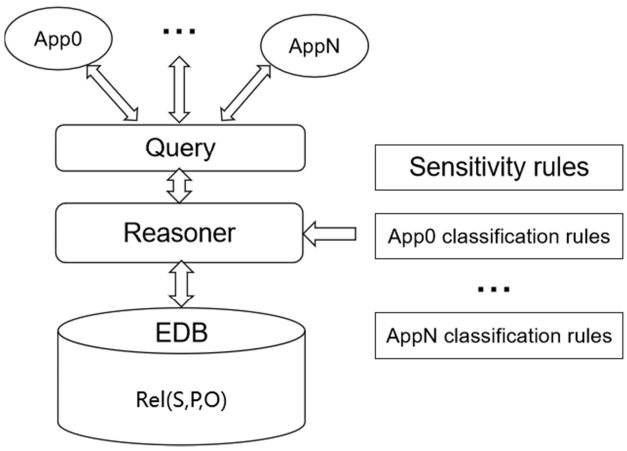

Figure 1. *The declarative framework.*

4. Sensitivity

Sensitivity is often implemented as a global value attached to data objects. For use in MAC, sensitivity forms a lattice $(\text{SensLevel}, \leq)$, in which SensLevel corresponds to the data type of sensitivity. A data object can only be read by users of equal or higher sensitivity levels, and be written by users of equal or lower sensitivity levels. Assuming a data object O is a target of sensitivity configuration, O may be a cell, a record, a column, a document, a database, etc. Sensitivity rules define sensitivity level of O, which is expressed as a Datalog literal.

$$\text{assign}(\text{O}, \text{L})$$

where L is in the set SensLevel. Because Datalog is an untyped language, a typing system may be added to the language or sensitivity rules can be specified in a way that L is guaranteed to be in SensLevel. Rules with assign() as the head literal define sensitivity levels of arbitrary data objects, whereas conventional label systems can only store labels in record-like entries.

The storage structure of data objects and the inference rules of sensitivity according to storage structures can be formalized in Datalog clauses. For example, one may expect that if a database is assigned sensitivity level L, then all the tables and

records in it will inherit this sensitivity level.

$$\text{assign}(X, L) \Leftarrow \text{assign}(Y, L), \text{storedin}(X, Y)$$

$$\text{storedin}(X, Z) \Leftarrow \text{storedin}(X, Y), \text{storedin}(Y, Z)$$

Note that it is possible that multiple sensitivity levels are assigned to the same data object. It is convenient to use Datalog queries to find such objects, and to resolve multiple assignments. For example, for a data object, one may want to assign the maximum level among all assignments, whereas for a user, to assign the minimum level.

$$\text{slevel}(O, \max\langle L\rangle) \Leftarrow \text{assign}(O, L)$$

$$\text{ulevel}(O, \min\langle L\rangle) \Leftarrow \text{assign}(O, L)$$

where slevel is used to define data sensitivity, and ulevel is to define user clearance. In this way a unique global sensitivity level can be define for any object. Again, a typing system may be added to confine the variables.

5. Classification

Compared to sensitivity, which has a rather limited application scenario, data classification enables more powerful ways of data protection. Data classification essentially only specifies the need of classifying data from multiple points of view, thus leaving space to implementing more sophisticated data security mechanisms. In practice, there are several ways to classify data.

The simplest way to classify data objects is based on column/attribute names. A hierarchy of classification can be built to divide columns into different groups. For example, the columns that store personal IDs can be classified as identity information, which belongs to social information. It is easy to express typical hierarchical classification in Datalog, using constraints including subclass and disjoint. Again, the classification type of a data object O in the simplest way can be expressed as a literal.

$$\text{classify}(O, T)$$

where T is in a classification set defined from some application's point of view. In contrast with sensitivity, multiple types can be attached to the same data object, so that applications can manage data independently. Moreover, it is possible to implement more complex classification utilizing the logic programming framework.

First, it is possible to define and compute classification that involves complex algorithms or models, including graph computing, dynamic programming, machine learning, as long as the algorithms or models satisfy Datalog's minimal fixpoint semantics. [10-13] For example, the classification type can be assigned to the lowest value of some attribute among all the interconnected data objects.

$$\text{class}(X, C) \Leftarrow \text{rel}(X, 'attr0', C)$$

$$\text{link}(X, Y) \Leftarrow \text{rel}(X, 'attr1', Y)$$

$$\text{link}(X, Y) \Leftarrow \text{rel}(Y, 'attr1', X)$$

$$\text{class}(X, C) \Leftarrow \text{link}(X, Y), \text{class}(Y, C)$$

$$\text{classify}(O, \min\langle C\rangle) \Leftarrow \text{class}(O, C)$$

Viewing the database storing triples in the form of (S, P, O) as a graph in which nodes are S and O connected by P = 'attr1', then above rules propagate the attribute values of 'attr0' among all nodes in a connected component and classify every node in a connected component as the same minimum 'attr0' value.

Second, it is possible to expand classification to incorporate parameters and other data objects. For example, still consider the above propagation problem, in practice super nodes are often found in a graph which are few but have a large number of edges. Super nodes deteriorate the performance of graph computation rapidly, so engineers often use some tricks to get around super nodes.

$$\text{degree}(X, \text{num}\langle N\rangle) \Leftarrow \text{link}(X, Y)$$

$$\text{class}(X, C) \Leftarrow \text{rel}(X, 'attr0', C)$$

$$\text{class}(X, N, C) \Leftarrow \text{link}(X, Y), \text{degree}(Y, M), M < N, class(Y, C)$$

$$\text{classify}(O, N, \min\langle C\rangle) \Leftarrow \text{class}(O, N, C)$$

where a parameter N is used as a threshold to prevent super nodes from propagating the attribute value. Removing super nodes in this way is an easy but ad hoc method, and therefore the final classification is also attached with N for clarity and ease of debugging.

In addition, it is also common to classify objects according to relations. For example, a common recommendation method in social networks is to find common friends. Two users are more likely to become friends if they share many common friends. Likewise, it is possible to classify two data objects as similar objects if they share a lot of attribute values.

$$\text{sameattr}(X, Y, P) \Leftarrow \text{rel}(X, P, V), \text{rel}(Y, P, V)$$

$$\text{sameattrcount}(X, Y, \text{num}\langle P\rangle) \Leftarrow \text{sameattr}(X, Y, P)$$

$$\text{classify}(X, Y, N, 'similarobjects') \Leftarrow \text{sameattrcount}(X, Y, M), M > N$$

where the classification is associated with a pair of data objects and a parameter. Applications can therefore design data protection mechanism based on relations and with parameters. For example, the derived $\text{classify}(X, Y, N, 'similarobjects')$ naturally constructs a partition of data objects, and similar objects within the same partition can be applied with the same protection strategy.

After defining sensitivity and classification, it is necessary to validate the results, which is also possibly done within the same framework by defining rules to find conflicts or unwanted results of sensitivity and classification given the test data objects.

6. Conclusion

A declarative method is proposed to realize data sensitivity and classification management. The correct definition and computation of sensitivity and classification is the base of further data protection mechanisms, such as access control [14], anonymization, and hypothesis test. [15] The declarative approach provides an abstraction layer that separate security requirements from actual implementations. The approach is therefore transparent to technical migrations and optimization. It is also easier for security experts to concentrate on security problems without considering the underlying big data systems and to leave the work of keeping equivalence to reasoning engine developers, who do not need to understand security.

References

[1] Bosong Liu, Research and implementation of security policy management system based on multidimensional attribute label, Master thesis, Beijing University of Posts and Telecommunications, 2017.

[2] Yuze Jiang and Shiyang Chen, Trends and challenges of data security technology, Communications World. 08 (2021): 17-19.

[3] Molham Aref, Balder ten Cate, Todd J. Green, Benny Kimelfeld, Dan Olteanu, Emir Pasalic, Todd L. Veldhuizen, Geoffrey Washburn, Design and Implementation of the LogicBlox System, SIGMOD'15, 2015.

[4] B. Motik; Y. Nenov; R. Piro, I. Horrocks, Parallel Materialisation of Datalog Programs in Main-Memory RDF Databases, In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, Qébec City, Qébec, Canada: 2014.

[5] Alexander Shkapsky, Mohan Yang, Matteo Interlandi, Hsuan Chiu, Tyson Condie, Carlo Zaniolo, Big Data Analytics with Datalog Queries on Spark, SIGMOD'16, San Francisco, CA, USA: 2016.

[6] Jiaqi Gu, Yugo Watanabe, William Mazza, Alexander Shkapsky, Mohan Yang, Ling Ding, Carlo Zaniolo, RaSQL: Greater Power and Performance for Big Data Analytics with Recursive-aggregate-SQL on Spark, SIGMOD'19, 2019.

[7] Grigoris Antoniou, Sotiris Batsakis, Raghava Mutharaju, Jeff Z. Pan, Guilin Qi, Ilias Tachmazidis, Jacopo Urbani and Zhangquan Zhou, A Survey of Large-Scale Reasoning on the Web of Data, The Knowledge Engineering Review, Vol. 33, 1–24, 2018.

[8] Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, Dan Olteanu, Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems, AAAI, 2014.

[9] Mohan Yang, Declarative Languages and Scalable Systems for Graph Analytics and Knowledge Discovery, PhD dissertation, University of California, Los Angeles, 2017.

[10] Carlo Zaniolo, Mohan Yang, Matteo Interlandi, Ariyam Das, Alexander Shkapsky, Tyson Condie. Fixpoint semantics and optimization of recursive Datalog programs with aggregates. TPLP 17 (5-6), 2017, 1048-1065.

[11] Ariyam Das, Youfu Li, Jin Wang, Mingda Li, Carlo Zaniolo. BigData Applications from Graph Analytics to Machine Learning by Aggregates in Recursion. Conference on Logic Programming (ICLP'19), 2019.

[12] Hongyuan Mei, Guanghui Qin, Minjie Xu, Jason Eisner, Neural Datalog through Time: Informed Temporal Modeling via Logical Specification, Proceedings of the 37th International Conference on Machine Learning, PMLR 119, Online: 2020.

[13] Wang, J., Wu, J., Li, M. et al. Formal semantics and high performance in declarative machine learning using Datalog. The VLDB Journal, 2021.

[14] Edelmira Pasarella, Jorge Lobo, A Datalog Framework for Modeling Relationship-based Access Control Policies, SACMAT '17, 2017, 91-102.

[15] Xinming Ou, A logic-programming approach to network security analysis, PhD dissertation, Princeton University, 2005.

Biography

Yuejin Zhang, PhD, Professor, graduated from Tsinghua University in 1998, research areas include computer science and technology, network security, etc.

Hong Liu, PhD, research area is cybersecurity.

Guowei Wang, MS, research area is cybersecurity.