Multi-Tenant Management and Research Based on the Big Data Platform of Sentry

Yuanyuan Chang, Meina Song, and Haihong E

***Abstract*—To realize authentication and access control of multi-tenant is one of the important problems to build big data platform. This thesis raises an access control policy based on the authentication mechanism of traditional Kerberos and the new incubation project of Apache, Apache Sentry. By combining Sentry and traditional protocols and the advantages of Sentry in the ecology of Hadoop, authentication and access control of big data platform are completed to guarantee the strong isolation among different tenants and moderate isolation among resources of different tenants.**

***Index Terms*—Multi-tenant, big data platform, Kerberos, apache sentry, Hadoop.**

I. Introduction

Now data safety problems of multi-tenant on big data platform are mostly solved by traditional access control strategy, which is improving the algorithm of Kerberos protocol [1]. For example, Yang Ping from China solved the problems of guessing attacks and complex storage of symmetric key by the mechanism of public key encryption and private key decryption in traditional Kerberos protocol [2]. She also added corresponding roles or groups for system based on traditional RBAC to improve access control model. As the demand for access control in big data increases, traditional model can not satisfy the new cloud computing architecture any more. Take RBAC [3] as an example. Definitions of subject and object in cloud change a lot and many service modes with the core of tenant and the basis of big data appear and they lead to optimization and upgrade of traditional access control model [4]. So, to better cooperate with the environment of big data in Hadoop ecology to realize access control, Cloudera released a Hadoop open source component, Apache Sentry, with the Fine-grained, authentication based on role and multi-tenant management mode to provide unified access control for data and metadata stored in Hadoop. Now Sentry can be integrated with Hive/HCatalog, Apache Solr and Cloudera Impala. Users can store more sensitive data in Hadoop. More terminal users can access to data in Hadoop.

CAD big data platform is self-developed big data storage and processing platform and built based on Hadoop with functions including data storage and processing, authentication, access control and data monitoring [5]. This platform use Kerberos for authentication and Sentry for access control to ensure data safety under multi-tenant.

Manuscript received October 23, 2017; revised December 20, 2017.

Yuanyuan Chang, Meina Song, and Haihong E. are with the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China (e-mail: cyybupt@126.com, {mnsong, ehaihong}@bupt.edu.cn).

II. Authentication – Kerberos Protocol

Authentication identifies users by their identifications to prevent attacker from counterfeiting legitimate user to obtain access right. For general computer network, authentication of host and node is considered and authentication of user can be realized by application system [6].

As a distributed authentication service system, Kerberos is an important authentication protocol for the Internet. It is mainly applied for network authentication. By inputting authentication information for once, users can get SSO (Single Sign On) [7], ticket used in accessing multiple services. Its principle structure is shown in following Fig. 1:

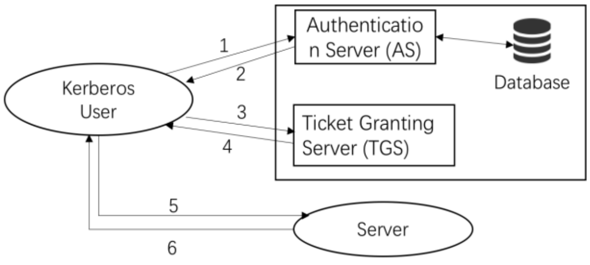


Fig. 1. Kerberos protocol structure.

Authentication processes: (1) User requests a ticket and a session key which can communicate with TGS from AS of Kerberos. (2) AS verifies if the only identification of the user is stored in database. If so, AS will generate a Client-TGS session and encrypt with it. It will also generate a ticket-granting ticket (TGT) for user to access TGS. (3) After receiving information returned by AS, user can decrypt and get a session key used to communicate with TGS and send ticket-granting ticket and access request information for authentication to TGS. (4) TGS uses information received by session key to verify the user’s identification by decrypted timestamp. (5) User decrypts information sent by TGS, gets a session key used to communicate with server and submits service granting ticket (ST) and self-generated authentication information to remote server. (6) Server compares information and authentication information in ST sent by user. After getting confirming information from application system, user decrypts the timestamp and verifies its legitimacy for authentication of the server [8].

Kerberos is based on symmetric cipher so it needs support from online AS which is responsible for distribution and management of keys, thus lightening the load of server. Ticket can be reused within validity term thus reducing using frequency of user password. Timestamp in authentication information is used to resist replay attack.

DOI: 10.7763/IJCTE.2017.V9.1181

Temporary session key is introduced as random factor which makes the system safer and realizes the authentication of the platform.

III. Access Control - Apache Sentry

Aim of Apache Sentry is to realize authorization management. It is a policy engine used by data processing tool to verify access permission [9]. It is also a highly scalable module that can support any data model. Now it supports the relation data models of Apache Hive and Cloudera Impala and data model with inheritance of Apache.

A. Components and Architecture

Components of Sentry are shown in Fig. 2

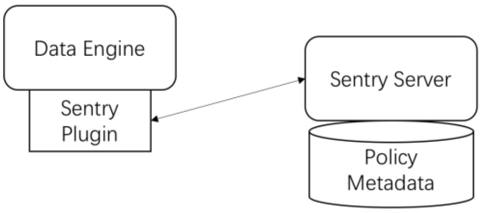


Fig. 2. Components of sentry.

Sentry server manages management authorization metadata and supports sage retrieval and manipulation of interface of metadata. When data processing application, i.e. data engine needs to use plug-in of Sentry, all clients accessing resources are requested to be blocked and sent to plug-in of Sentry for authentication. Sentry persists the mapping of roles, permission and role combination to a relation database and provides programmed API interface for creating, inquiring, updating and deleting. It also allows its clients to obtain and modify permissions.

Architecture of Sentry is shown in Fig. 3:

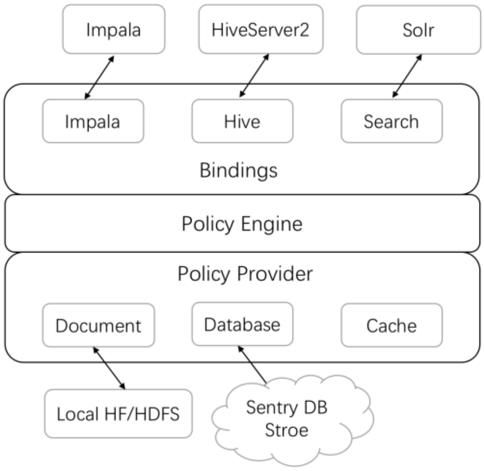


Fig. 3. Architecture of sentry.

There are three important components in the architecture of Sentry: Binding, Policy Engine and Policy Provider.

Binding can provide authorization to different query engines while inserting its Hook function into phases of compiling and execution of SQL engines. Two roles of the Hook functions are: 1) working as filter to pass SQL query with access right to corresponding data object and 2) permission takeover. After using Sentry, permission of grant/revoke is taken over by Sentry and the execution is realized in Sentry. All engine permission information is stored in the unified database set up by Sentry. Centralized management of engine permission is realized.

Policy Engine is the core component of Sentry. It compares permission demand input from Binding with stored permission description to see if they are matched.

Policy Provider is responsible for reading set access permission from document or database.

Sentry has following key concepts:

Verification — reliable identification of the user's authentication credentials;

Authorization — limiting user access to given resource;

Users — individuals identified by the underlying certification system;

Group — a group of users maintained by the certification system;

Permission — authorization rules or instructions to access a particular resource;

Role — a series of permissions, one or more templates of access rules

Sentry defines the object which needs to comply with authorization rules and the granularity of the allowed action. For example, in the SQL model, the object can be a database or table, the operation is SELECT, INSERT, CREATE and so on. For search models, objects are indexes, collections, and documents; access patterns are queries, updates, etc. Sentry associates users with groups, puts a series of users into a group, but Sentry can not give a user or group license directly, You need to delegate permissions to roles, and roles can be delegated to a group instead of a user.

B. Combined with Hadoop

Apache Sentry can be used with Hadoop components. Integration of Sentry and Hadoop is shown in following Fig. 4:

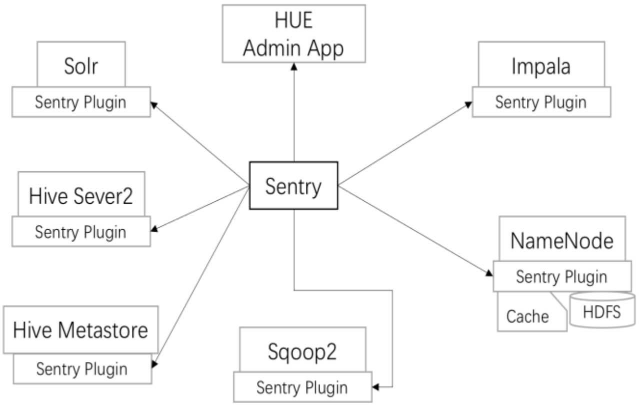


Fig. 4. Integration of sentry and hadoop.

Its core service is Sentry server which stores authorized metadata and provides API, the tool for safe retrieval and modification of the metadata. Sentry server can be only used for storage and basic operation of metadata. The actual permission is conducted by policy engine which operates in data processing applications like Hive or Impala. All components in Hadoop can request to load plug-ins of Sentry including service client processing Sentry service and policy engine verifying permission request.

This platform mainly uses the combination of Sentry and HDFS. Sentry-HDFS permission focuses on warehouse data of Hive, i.e. any data in a part of Hive or Impala form. The integration aims to expand the same authorization check to access warehouse data of Hive through any other components such as Pig, MapReduce or Spark.

Permissions of Sentry mapped to HDFS ACL [10] are as follows:

SELECT —> permission of reading files;
INSERT —> permission of writing files;
ALL —> permissions of reading and writing files.

IV. Access Control Policy of CAD Big Data

CAD big data platform as a multi-tenant platform needs to meet requirements of different enterprises for safety and guarantee data isolation among enterprises and data isolation and cooperation among departments. So the access control policy raised in this thesis is based on RABC in Sentry components. By the mapping method of permission – role – user group – user in the authorization model of Sentry, enterprise can create corresponding role in accordance with user's label. Only roles have different permissions. After the role of user or user group is created, user can have relevant permissions which ensures the isolated, shared and safe data access.

A. Definition of Role

Following method are provided to define roles of users in enterprise on CAD big data platform: after enterprise user registers on the platform, it shall select a role from several safety property tags provided by the platform based on the situation of the enterprise, including department, center and project group. Project group can have sub-project group. For example, after a user attached to Project Group A, Center A, Department A registers on the platform and selects names of enterprise, department, center and project group, access permission of the user is settled, i.e. all workspace of Project Group A, Center A, Department A of the enterprise. Role tags in levels make the enterprise structure in tree form and role control clearer. This thesis uses role concept defined by Sentry, combination of a series of permissions and one or more templates of access rules. The definition is isolated from concepts of user and user group. User has following roles in access control policy of CAD big data platform:

- Data Administrator: Data Administrator has administrative permission to data of all tenants in the whole system. He can partition workspaces and data for enterprise dynamically in accordance with requirements of the enterprise and synchronize data of the tenants and monitors logs.
- System Administrator: System Administrator's permission is lower than Data Administrator, He can operate data in the data space of all platforms the enterprise owns, such as data modeling and sample data analyzing. He can also set up Application Administrators for workspace of each level. However, he cannot check data information of tenants on the whole platform and cannot change his own role.
- Application Administrator: Application Administrator is appointed by departments under centers in the enterprise. He can only access data platform through API and cannot operate and access source data. He can add users in his workspace and modifies his users' permissions.
- User: User has the lowest permission in each workspace. He can check available operation in the workspace but cannot modify. He can only access data in space authorized by Application Administrator.

Different form the traditional one fixed permission for one role, the access control policy based on role and Sentry raised in this thesis depends on basic Kerberos authentication system. It uses the group mapping mechanism configured in Hadoop to ensure other components in the system of Sentry and Hadoop complying with the same group mapping.

B. Access Control Process

Access control is to ensure that user and application have proper and relevant permission to use proper data set and metadata and simplify management and control process by role-form authorization policy. Sentry can use Active Directory (AD) to decide user group distribution by defining and establishing roles. If anything changes, it can also upgrade the role distribution. In addition, policy adding or changing a user for functional roles can be carried out by a simple order of SQL or HUE UI.

For example, give User A, B and C different roles. A is System Administrator and Application Administrator, B is Application Administrator and C is User. Take Application Administrator as example. Declare available operations for the role:

```
    # group to role mapping
    [groups]
    A = System_Administrator_role,
Application_Administrator_role
    B = Application_Administrator_role
    C = User_role
    # role to privilege mapping
    [roles]
    Application_Administrator_role =
server=server1->db=customer_info, \
    server=server1->db=goods->table=*-
>action=select, \
   server=server1->db=default->table=tab2
```

V. System Test

Laboratory environment sets up Hadoop cluster for the six computers in Linux system. Hadoop is 2.7.3 version with 64GB memory for each node, 24 cores and developed by vue.js at front end.

CAD big data platform divides current resources at Hadoop cluster in two part, one for development and usage for members in lab and the other for experiment and teaching for teachers. So there are two centers on the platform, one CAD Core and one Teach. Core consists of design, front end, development and test departments. Teach consists of case, development and test departments. Each department has different groups and users as shown below Fig. 5:

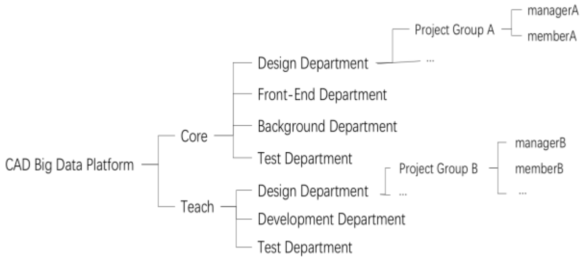
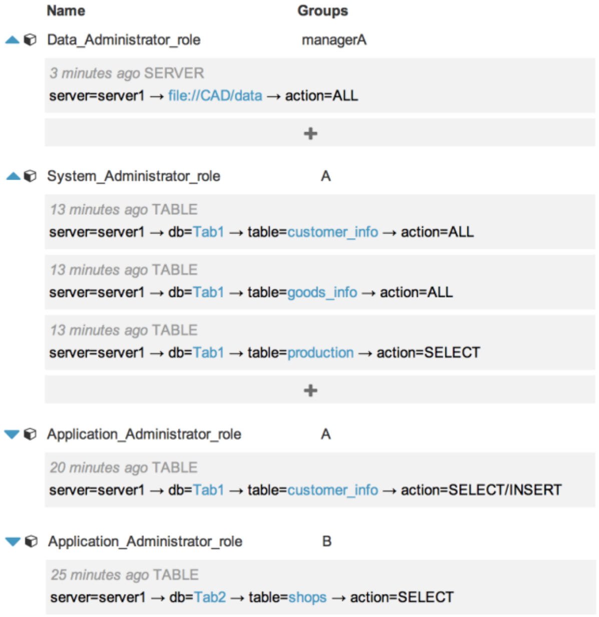


Fig. 5. Tree architecture of CAD big data platform.

Grant each role with different permissions. The screenshot of authorization process on front end interface is as follows Fig. 6:



Name	Groups
Data_Administrator_role	managerA
System_Administrator_role	A
Application_Administrator_role	A
Application_Administrator_role	B

Data_Administrator_role — managerA

3 minutes ago SERVER
server=server1 → file://CAD/data → action=ALL

System_Administrator_role — A

13 minutes ago TABLE
server=server1 → db=Tab1 → table=customer_info → action=ALL

13 minutes ago TABLE
server=server1 → db=Tab1 → table=goods_info → action=ALL

13 minutes ago TABLE
server=server1 → db=Tab1 → table=production → action=SELECT

Application_Administrator_role — A

20 minutes ago TABLE
server=server1 → db=Tab1 → table=customer_info → action=SELECT/INSERT

Application_Administrator_role — B

25 minutes ago TABLE
server=server1 → db=Tab2 → table=shops → action=SELECT

Fig. 6. Role access control based on sentry.

As shown by above assignment of user's role by SQL statement, it is very convenient to use visual interfaces of Sentry components to assign roles by adding role and permission to user. Permission, role and user group all are assigned by SQL statement of grant/revoke. "User group" and influencing "user" are realized by the mapping of user-group of Hadoop. Hadoop provides two kinds of mappings: one os mapping from Linux/Unix user of local server to group and the other is mapping from user realized by LDAP [11] to group. The latter one is more appropriate for big system because it has centralized configuration and is easy to modify.

User A and B belong to different projects groups in different departments. So even if they are Application Administrators, they have different permissions. Data can only be accessed by user. User is strictly authenticated each time accessing the data which controls strong data isolation among enterprises effectively. Among project groups, centers and departments, data shall be shared while be isolated. Sentry expands the data objectives supported by traditional RABC from database/form/view to server, URI and column size level, which brings great convenience to access control.

VI. Conclusion

Multi-tenant access control policy based on Sentry is combined with Kerberos authentication mechanism, shows combined advantage with components in Hadoop and ensures the safety, flexibility, hierarchy and high efficiency of big data platform. After tested, Sentry has the ability of controlling the data of authenticated user at Hadoop cluster and executing level permission precisely. It improves the interactive performance of permission management of big data platform and shows reliable data isolation and modularization. Then, we will begin to study the simplification of the deployment and management of Sentry's rights, and to extend the permissions control support for mature relational databases. We will study the problem of Sentry's control of next-generation rights access control, such as the tag-level control.

Acknowledgment

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China (Grant No. 2015BAH07F01); the Special Found for Beijing Common Construction Project; Engineering Research Center of Information Networks, Ministry of Education.

References

[1] B. Tung, "Public key cryptography for initial authentication in Kerberos," *Work in Progress*, pp. 72–79, 2006.

[2] P. Yang and H. Y. Ning, "Study on security analysis and countermeasures of Kerberos protocol," *Computer Engineering*, vol. 41, no. 5, pp. 144-148, 2015.

[3] S. Jiang, "Role-based access control models," *Computer Science*, vol. 4, no. 3, pp. 190-251, 2009.

[4] Y. L. Wang and J. Wang, "A access control strategy for data Storage platform based on Kerberos and HDFS," *Software*, vol. 37, no. 1, pp. 67-70, 2016.

[5] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Communications of the ACM*, vol. 33, no. 2, pp. 168-176, 2008.

[6] J. K. Li, D. Y. Zhang, Y. Zhang, "Research on identity authentication mechanism and its security analysis," *Application Research of Computers*, vol. 18, no. 2, pp. 126-128, 2001.

[7] A. A. Pirzada, "Mcdonald C. Kerberos Assisted Authentication in Mobile Ad-hoc Networks," in *Proc. Australasian Conference on Computer Science*, Australian Computer Society, Inc., 2004.

[8] L. J. Pu, *Research and Implementation of Unified Identity Authentication and Authorization Management System Based on Kerberos and LDAP Protocol*, Beijing University of Posts and Telecommunications, 2015.

[9] M. Backes, I. Cervesato, A. D. Jaggard, *et al*., "Cryptographically sound security proofs for basic and public-key Kerberos," *International Journal of Information Security*, vol. 10, no. 2, pp. 107-134, 2011.

[10] A. V. Bechtolsheim and D. R. Cheriton, *Access Control List Processing in Hardware*, US, US7023853 [P], 2006.

[11] V. Koutsonikola and A. Vakali, "LDAP: Framework, practices, and trends," *IEEE Internet Computing*, vol. 8, no. 5, pp. 66-72, 2004.

Yuanyuan Chang was born in Shanxi, China, now he is studying at the Beijing University of Posts and Telecommunications as a master student, majoring in computer science and technology. Her research interests include cloud computing and large data, data visualization, font end, etc.

Meina Song graduated from Beijing University of Posts and Telecommunications in 2004 with a Ph.D degree. Her research interests include service computing, cloud computing and large data, machine learning and in-depth learning, smart city, modern service industry, etc.

Prof. Song's publications are Statistic-based CRM approach via time series segmenting RFM on large scale data; Mu-En: Multi-path of entity recommendation based on path similarity; A link prediction algorithm that solves the data sparsity problem in service QoS prediction; Cross-layer power allocation scheme for cellular network using cooperative diversity; A customizable asset management system architecture based on extended-SOA, etc.

Haihong E graduated from Beijing University of Posts and Telecommunications in 2010 with a Ph.D. Her research interests include cloud computing and large data, machine learning and depth learning, data visualization, service computing, mobile Internet, etc.

Asst. Prof. E has some publications: Incremental weighted bipartite algorithm for large-scale recommendation systems; Measure method and metrics for network characteristics in service systems; A general rating recommended weight-aware model for recommendation system; Dynamic scheduling of workflow for makespan and robustness improvement in the iaas cloud, etc.