Poster: A Pluggable Authentication Module for Big Data Federation Architecture

Feras M. Awaysheh, José C. Cabaleiro,
Tomás F. Pena
{feras.awaysheh,jc.cabaleiro,tf.pena}@usc.es
Centro de Investigación en Tecnoloxías da Información
Universidade de Santiago de Compostela, Spain

Mamoun Alazab
alazab.m@ieee.org
College of Engineering, IT, and Environment
Charles Darwin University, Australia

ABSTRACT

This paper intends to propose a trustworthy model for authenticating users and services over a Big Data Federation deployment architecture. The main goal of this model is to provide a Single-Sign-on (SSO) approach for the latest Hadoop 3.x platform. To achieve this, a conceptual model is proposed combining Hadoop access control primitives and the Apache Knox framework. The paper provides various insights regarding the latest ongoing developments and open challenges in this domain.

CCS CONCEPTS

• **Information systems → Data access methods**; • **Security and privacy → File system security**; • **Software and its engineering → Abstraction, modeling and modularity**.

KEYWORDS

Big Data, HDFS federation, Authentication, SSO Reference Model

ACM Reference Format:
Feras M. Awaysheh, José C. Cabaleiro, Tomás F. Pena and Mamoun Alazab. 2019. Poster: A Pluggable Authentication Module for Big Data Federation Architecture. In *The 24th ACM Symposium on Access Control Models and Technologies (SACMAT '19), June 3–6, 2019, Toronto, ON, Canada.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3322431.3326448

1 INTRODUCTION

Apache Hadoop, the prominent Big Data (BD) paradigm, has become a large-scale data analytic operating system. The large community behind Hadoop have been working to improve its ecosystem to meet the security increasing demands and requirements. The new features shift proves the Hadoop 3.x maturity and applicability to serve different markets. Enterprises across all major industries have adopted Hadoop for its capability to store and process an abundance of new types of data and leverage modern data architecture. With a broad spectrum of both structured and unstructured workloads, Hadoop abstracts the computing resource management, task scheduling, and data management, while maintaining a satisfying level of security and isolation.

The Hadoop Distributed File System (HDFS) is typically deployed as part of a large-scale Hadoop platform, to supports low-cost commodity hardware and accommodate different processing frameworks. It is utilized to handle data management and access to the Apache Hadoop ecosystem, using a master/slave architecture. The latest advancement in BD platforms leads to support of a clear separation of data management and its physical storage. Using multi namespaces in the form of federation architecture, Hadoop improves its scalability and isolation.

In this paper, we propose a mechanism to integrate Hadoop 3.x authentication schemes into Apache Knox [3] in a high-level abstraction. This presents a first work to define and formalize a service gateway for a BD federation (BDF) deployment architecture. The key question we will be asking for addressing the access control challenge is "How to design SSO abstract that handles Data/Service access in a Hadoop federation?" We address this question by defining a SSO reference architecture (SSORA) for the secure development of BDF. The work presented in this paper may employ in a Hadoop federation environment across multi-tenant BDaaS and IaaS clouds, as well as on-premises datacenters.

2 HDFS FEDERATION

HDFS [7] is composed of two primary daemons, (i) a single NameNode (NN) that is deployed at the cluster master node, and (ii) several DataNodes (DNs) running at the cluster slaves (usually one per-node). The NN runs the namespace process, which manages the file system information and regulates access to files using a traditional hierarchical organization.

To enable universal block storage layer, Hadoop performed separation of namespace and blocked storage [2]. A federation BD environment, through multi-independent namespaces for block management and a shared block pool for data storage, improves scalability and isolation of Hadoop operations. So, by losing the tightly-coupled block storage and namespace, each DN registers with all the NNs in the cluster (raises authentication requirement). This allows to scale the NN horizontally and enable the aggregation of geo-distributed Hadoop clusters. This feature directly enhances throughput by adding more access enforcers (typically NNs in HDFS architecture), which improve read/write operations.

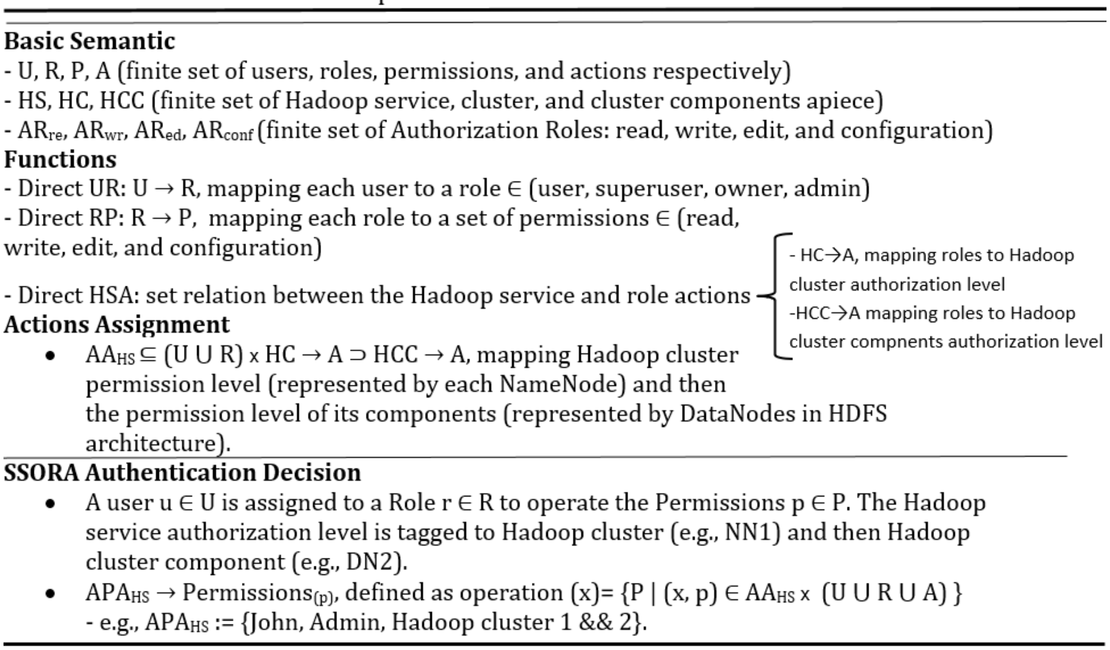
3 SSORA ACCESS PATTERN

There is a necessity to create a common language tailored to build an efficient, elastic and autonomous access control reference model for BDF. This reference model will not only define patterns that can result in the dynamic optimization of accelerating access enforcement, but it will keep access discussions to a minimum as well. Here,

SACMAT '19, June 3–6, 2019, Toronto, ON, Canada

ACM ISBN 978-1-4503-6753-0/19/06.
https://doi.org/10.1145/3322431.3326448

Table 1. Hadoop Federation SSORA Model Definitions.



Basic Semantic

- U, R, P, A (finite set of users, roles, permissions, and actions respectively)
- HS, HC, HCC (finite set of Hadoop service, cluster, and cluster components apiece)
- AR_{re}, AR_{wr}, AR_{ed}, AR_{conf} (finite set of Authorization Roles: read, write, edit, and configuration)

Functions

- Direct UR: U → R, mapping each user to a role ∈ (user, superuser, owner, admin)
- Direct RP: R → P, mapping each role to a set of permissions ∈ (read, write, edit, and configuration)
- Direct HSA: set relation between the Hadoop service and role actions
  - HC→A, mapping roles to Hadoop cluster authorization level
  - HCC→A mapping roles to Hadoop cluster compnents authorization level

Actions Assignment

- $AA_{HS} \subseteq (U \cup R) \times HC \rightarrow A \supset HCC \rightarrow A$, mapping Hadoop cluster permission level (represented by each NameNode) and then the permission level of its components (represented by DataNodes in HDFS architecture).

SSORA Authentication Decision

- A user u ∈ U is assigned to a Role r ∈ R to operate the Permissions p ∈ P. The Hadoop service authorization level is tagged to Hadoop cluster (e.g., NN1) and then Hadoop cluster component (e.g., DN2).
- $APA_{HS} \rightarrow Permissions_{(p)}$, defined as operation (x)= {P | (x, p) ∈ AA_{HS} x (U ∪ R ∪ A) }
  - e.g., APA_{HS} := {John, Admin, Hadoop cluster 1 && 2}.

we present a HDFS federation SSO Reference Architecture (SSORA), a sequential pattern-based access control for a federated SSO model. This will aid in clarifying the required tools/mechanisms to implement a SSO architecture for Hadoop 3.x clusters.

SSORA affords a binding among the external clients and the Hadoop services in a federated Hadoop platform. It enables the internal entities (e.g., NN and DN daemons) to verify these bindings through the use of a central policy authorization and several distributed policy agents. We will now discuss the formal definitions of SSORA model as shown in Figure 1 and specified in Table 1.

`SSORA Pluggable Module` enables fast integration with architecture (i.e., allows to plug functionality without modification), it does not require making any modification of the prior cluster design. Before addressing the SSORA and federation access pattern, there are few definitions to guide and facilitate the discussion:

- Apache Knox service is utilized as a unified access point of the federation service.
- Hadoop daemons and services (internal communication) are authenticated using Hadoop core capabilities, i.e., Kerberos principals and their access granularity is managed using POSIX ACLs.
- Knox authenticates clients (admins may set an LDAP group), and Apache Ranger [4] applies their access granularity.

In a federation environment, it is often the case that a client of a particular HDFS cluster (e.g., NN1) wishes to utilize services offered by another administrative domain (e.g., NN2). In such cases, the user needs to be authorized by the central policy authority (CPA) to reach the remote domain. The necessary steps required for a given BDF to access the provided SSO federation are described subsequently (see Figure 1):

(1) The Client asks to access the Hadoop cluster, interface with the cluster gateway (Knox) using his identity. If the external client wish to upload new data to the Hadoop data lake, the gateway may ask to secure the connection via SSL/TLS communications over the public network.
(2) If Knox cannot find a valid session for the access request (for first-time registration) it start a secure session with the CPA (ACL in Hadoop) on clients' behalf.
(3) The CPA validates the proof of authentication and issues a new security certificate to initialize his permissions (get the authorization level access). The certificate is forwarded to the gateway interface and performs a login if needed.
(4) Access certificate containing the ClientID is issued for valid users and is signed with the CPA private key. The client receives the certificate and adds it as an embedded proof of access. This certificate provides an abstraction for combining any number of authentication and authorization systems.

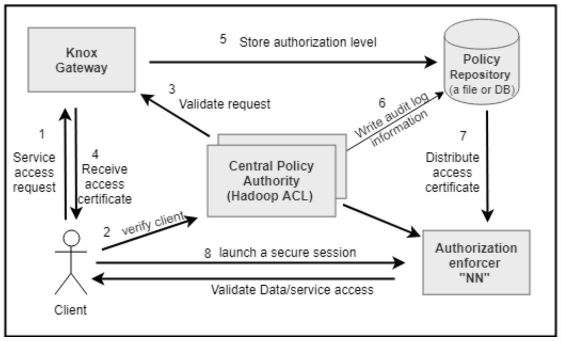

Knox Gateway
5 Store authorization level
Policy Repository (a file or DB)
3 Validate request
6 Write audit log information
1 Service access request
4 Receive access certificate
Central Policy Authority (Hadoop ACL)
7 Distribute access certificate
2 verify client
8 launch a secure session
Authorization enforcer "NN"
Validate Data/service access
Client

Figure 1: SSO Reference Architecture (SSORA) and federation access pattern.

Also, it accelerates authentication to a vast number of participants (under the federation umbrella).

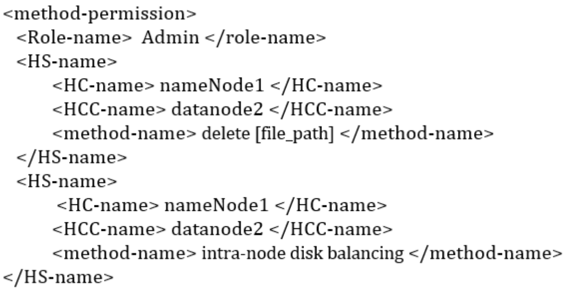
(5) The policy repository is updated, and the authorization level is stored. The CPA can always issue new security session without the need for storing the client identities, permissions, and logs locally. Policy agent will keep listening to the new authorization session.

(6) For auditing purposes and security analysis, the CPA manage user activities (at the service level) by writing audit logs in a common repository.

(7) The CPA centralizes the access control and shares the allowed user/calls via a push mechanism to the Authorization Enforcer (AE), which is, typically, the NN in the architecture.

(8) The trust relationship between the client and AE is established, and his requests are authorized to proceed. Other security layers may apply (e.g., an internal TLS and files encryption).

Figure 2 shows a sub-method of the admin permission in SSORA as an XML fragment policy that performs two operations (delete and intra-node balancing) over BDF.



```
<method-permission>
  <Role-name>  Admin </role-name>
  <HS-name>
       <HC-name> nameNode1 </HC-name>
       <HCC-name> datanode2 </HCC-name>
       <method-name> delete [file_path] </method-name>
  </HS-name>
  <HS-name>
        <HC-name> nameNode1 </HC-name>
       <HCC-name> datanode2 </HCC-name>
       <method-name> intra-node disk balancing </method-name>
</HS-name>
```

Figure 2: A representative SSORA XML example.

4 DISCUSSION: SUMMARY AND OPEN CHALLENGES

This work intends to propose a trustworthy and Federation-oriented model for authenticating users and services over a Hadoop 3.x. platform. The main goal of this model is to provide a SSO gateway based on Hadoop access control primitives and Apache Knox. A SSO architecture can be managed using Kerberos with Lightweight Directory Access Protocol (LDAP), which is enabled by setting `Hadoop.security.group.mapping.ldap.url` to true. Alternatively, implementing a single gateway that handles client access management behind a firewall can be achieved by employing Apache Knox. Knox will allow/deny users to access the ecosystem services before interacting with the Hadoop cluster. Following to user verification using Knox gateway, Knox will use its Kerberos principals to confirm, securely, with other Hadoop services and daemons.

It is, also, expected to adopt the SSORA within our BD opportunistic and elastic resource allocation (BigOPERA) platform. BigOPERA architecture (its prototype proposed at [5]) combines the computing power of available (non-dedicated) high-throughput resources to Hadoop 3.x dedicated cluster using Docker containers as worker nodes.

Examples of different open challenges and research directions include:

Scalable Authorization Despite the availability of several adapted security mechanisms and techniques that have been developed for BD security, more research activities are still needed to establish scalable security models and paradigms that must be driven by BD specifications and requirements. Also, new security abstractions for BDF are still needed to simplify the task of identifying the unimproved gaps. Also, leverage recent mechanisms of policies enforcement [1] to combine BDF with ABAC [6] could label as future research.

Fine-grained Authentication Management The need to develop active and dynamic models (policies and standards) for BDF is inevitable. The challenge here is to extend the previous approaches with new policy editing and presentation tools for flexible and extensible data access, i.e., policies that extend ABAC with stateful access sessions and mutable attributes (i.e., characteristics that change dynamically during the session). This extension is mainly invaluable to advance the authorization mechanisms of the multi-tenant data lake architecture as well as the latest BDF model. Tackling this issue can be achieved by deploying policy templates that extend the primary ACL (or any access decision enforcer) as a specialization of the XACML V3.0 standard.

Data-centric security This requires restricting the BDF access to only those authorized by succinct gateway. However, the current state-of-the-art BD-based encryption technologies are around transparent data encryption. This takes place at the file-level (for data at rest) and does not protect data-in-transit or data-in-process (not even the metadata). Attribute-based Encryption (ABE) in conjunction with usage control approach and ABAC can be labeled as a future direction for this issue. It could provide a complete end to end encryption, i.e., including data traffic between the store and application.

Providing new solutions in all those research directions will promote innovative BDF solutions with advanced security features.

ACKNOWLEDGMENTS

Spanish project TIN2016-76373-P, Xunta de Galicia accreditation 2016-2019, ED431G/08 and reference competitive group 2019-2021, ED431C 2018/19, and European Regional Development Fund (ERDF).

REFERENCES

[1] T. Ahmad, U. Morelli, S. Ranise, and N. Zannone. 2018. A Lazy Approach to Access Control as a Service (ACaaS) for IoT: An AWS Case Study. In *Procs. of the 23nd ACM on Symp. on Access Control Models and Technologies*. ACM, 235–246.

[2] Apache Hadoop. 2019. Apache Hadoop 3.x HDFS Federation Features. http://hadoop.apache.org/docs/r3.2.0/hadoop-project-dist/hadoop-hdfs/Federation.html. (2019). Accessed: 2019-02-10.

[3] Apache Knox. 2019. REST API and Application Gateway for the Apache Hadoop Ecosystem. https://knox.apache.org/. (2019). Accessed: 2019-02-10.

[4] Apache Ranger. 2019. Security management for the Apache Hadoop Ecosystem. https://ranger.apache.org/. (2019). Accessed: 2019-02-10.

[5] Feras M Awaysheh, Tomás F Pena, and José Carlos Cabaleiro. 2017. EME: An Automated, Elastic and Efficient Prototype for Provisioning Hadoop Clusters On-demand.. In *The 7th International Conference on Cloud Computing and Services Science, CLOSER*. 709–714.

[6] M. Gupta, F. Patwa, and R. Sandhu. 2018. An Attribute-Based Access Control Model for Secure Big Data Processing in Hadoop Ecosystem. In *Procs. of the Third ACM Workshop on Attribute-Based Access Control*. ACM, 13–24.

[7] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. 2010. The Hadoop Distributed File Dystem. In *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*. Ieee, 1–10.