ROLE-BASED AUTHORIZATION IN EQUIPMENT ACCESS AT CERN

K. Kostro, W. Gajewski, CERN, Geneva, Switzerland
S. Gysin, Fermilab, Illinois, U.S.A.

Abstract

Given the significant dangers of LHC operations, Role-Based Access Control (RBAC) is designed to protect from accidental and unauthorized access to the LHC and injector equipment. Role-Based Authorization is part of this approach. It has been implemented in the Controls Middleware (CMW) infrastructure so that access to equipment can be restricted according to Access Rules defined jointly by the equipment and operation groups. This paper describes the authorization mechanism, the definition and management of Access Rules and the implementation of this mechanism within the CMW.

MOTIVATION

Operation of LHC, given the high amount of stored energy, needs particular protection against unintended access [1]. Although people and sensitive equipment are already protected by hardware and software interlock systems, erroneous or malicious equipment access can still result in significant damage or machine downtime. There is also a need to define, between the machine operation and equipment groups, *who can do what and when*. These needs are not restricted to the LHC and in the new CERN controls environment, largely based on Java GUIs and generic tools, the possibilities for an erroneous manipulation, intended or not, are much bigger than before.

CONTROLS MIDDLEWARE

Controls Middleware is a software infrastructure delivered and managed by CERN Accelerators and Beams/Controls group. Its goal is to provide a generic way of accessing any accelerator devices in the LHC or its injectors.

Accelerator devices are represented using the CMW device/property model [3]. This object-oriented abstraction schema offers a concept of a *named accelerator device*. Such an entity is an instance of a given *device class* which represents either an actual physical device (e.g. Beam Position Monitor) or a virtual device (e.g. Beam Line). Each device class strictly defines a set of *properties* which can be accessed remotely to perform read/write operations (in the language of the CMW referred to as *get/set operations*). Also, the CMW offers a possibility for a distant client to *subscribe* (“monitor on” operation) for a certain property and to receive a notification upon its modification.

From the point of view of the CMW, each accelerator device is a CMW *server*. Applications in the CERN Control Centre (CCC) maintaining communication with accelerator devices are viewed as CMW *clients*.

Previously described functionality, central to the CMW, is provided in the form of the *Remote Device Access* (RDA) library [4]. The library is provided in Java as well as C++ versions. For C++, amongst the supported platforms is LynxOS and Linux.

AUTHORIZATION, AUTHENTICATION AND ROLES

Authorization (A2) is the process of verifying that a known person has the right to perform a certain operation. Prior to authorization, an operator starting a client application needs to be *authenticated* (A1). The authentication process is described elsewhere [2] and we assume that it has produced an *authentication token*, containing data needed for the authorization.

Since it is not practical to describe authorization restrictions for every single user, we group users into *roles*. Roles correspond to the actual role played by a group of users in the control system such as LHC Operator, Beam Transfer Expert, Beam Instrumentation Developer, etc. User can be member of several *roles*, according to different roles he might take, operating, maintaining or developing the control system.

Similarly, authorization permission is a function of *location* of the node issuing an operation. Location is represented by a set of host names and can be used to group e.g. all the computers from the CCC. For certain locations (e.g. from the CCC) all hosts are eligible for authorization by location, meaning that no user role is needed for an operation to succeed. However, critical operations should be protected by limiting the access only to selected user roles.

Correspondingly, an *application* issuing an operation can be subject to authorization. Some demands may be permitted only from a given application while for the others this argument is irrelevant.

Certain operations may be dependent on the *accelerator mode*, representing a phase in the work of the accelerator (e.g. SHUTDOWN or COOLING). Again, we expect the majority of permission rules to be independent of this argument.

SUBJECT OF AUTHORIZATION

Since virtually every equipment access is performed via CMW equipment servers, the RBAC authorization is based on the CMW device/property model. Every operation (get, set or subscription demand) is subject to the authorization process and its execution can be denied in case of issuer having insufficient privileges.

Other types of operations (e.g. reboot, configuration change) can also be controlled with use of the RBAC by introducing additional server properties and limiting their access with appropriate access rules.

ACCESS RULES

Decision whether a particular operation is valid or not is dependent on a set of *access rules*. They are specified by an *equipment specialist* for every device class, stored and managed centrally in the AB Controls database. Every CMW server can read access rules (referred to as *access map*), relative to device classes it is providing access to, through a tab-separated text file located in the AB/Controls Network File System. This file mirrors the access rules located centrally in the DB.

Access map is read by CMW server on its start-up. In addition, access map can be reread upon a distant call from the CMW client. It usually happens when the set of related access rules have been modified by the equipment specialists.

Access Rules Structure

Access rules are parameterized using all factors relative to the authorization process. More specifically, an equipment specialist has to specify the following fields to define an access rule:

- device class name,
- property name,
- device name,
- role name,
- application name,
- location name,
- accelerator mode,
- operation type (set, get, or subscribe).

Apart from specific values, an equipment specialist can put a wildcard ‘*’ in any of the fields except device class. This interpreted as ‘all values fit’.

Existence of a specific rule attributes access privileges to operations associated with given authorization parameters. A separate field assures that access can be differentiated based also on the type of operation. Typically, the *set* operation will be more restricted than the others as it is resulting in a change of the state of the underlying device.

The proposed structure of the access rules allows straightforward and natural definition of access patterns to devices. Furthermore, we expect an average access map to contain no more than 20-30 rules, which is an easily manageable number.

AUTHORIZATION PROCESS

Figure 1 presents the interactions between a CMW client, the CMW server and the RBAC server during an execution of a CMW server access. Firstly, upon start-up, the CMW server reads its access map from a file (1). Consequently, it is ready to receive calls from an application which is seen in our system as a CMW client. To allow access to the servers, the operator must authenticate himself [2] first and obtain a RBAC token from the RBAC server (2). The token is transferred to the CMW server in order to provide information needed for authorization (3, 4). Now CMW client’s access requests can be sent to the CMW server (5). To properly authorize a request, the CMW server obtains current accelerator mode from the timing source (6). Finally, authorization decision is taken and in case of access denial an RBAC exception is handled back to the CMW client (7).

Result of every authorization process, both positive and negative, is logged at a Log4J server (8). This allows easy diagnosing of access problems as well as auditing the complete access history.

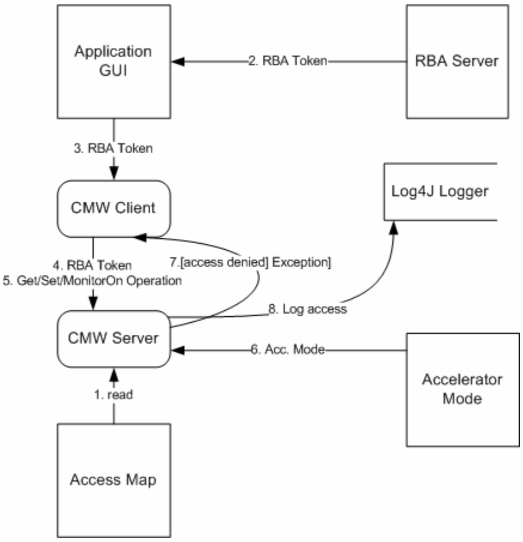

Figure 1: Data flow during CMW server access.

RBAC Token

RBAC token is an entity transferring information about authenticated user between RBAC server and CMW client [2] and subsequently to CMW server. It contains, among other, the following fields:

- token serial ID,
- authentication time,
- token expiration time,
- application name,
- location name,
- array of operator’s roles,
- token digital signature (SHA1 + RSA with 512 bit key).

Token digital signature is included to guarantee the authenticity of the token. The token is generated by the RBAC server and the signature certifies that it arrived unchanged at the CMW server.

There are two modes how CMW operations can be authorized:

- ‘token per connection’ and
- ‘token per operation’.

‘Token per connection’ is used by most of the 2-tier applications. A token is transmitted to the CMW server upon the connection establishment and is stored there. It is subsequently used to authorize every CMW operation. New remote call has been added to the CMW server to allow replacing ‘per connection’ token with another one.

‘Token per operation’ mode is used by 3-tier operator applications. Here, the RBAC token is sent encapsulated in the context of every CMW operation. This approach is heavier than ‘token per connection’ but allows a flexible switching between credentials used to authorize subsequent operations.

Authorization Algorithm

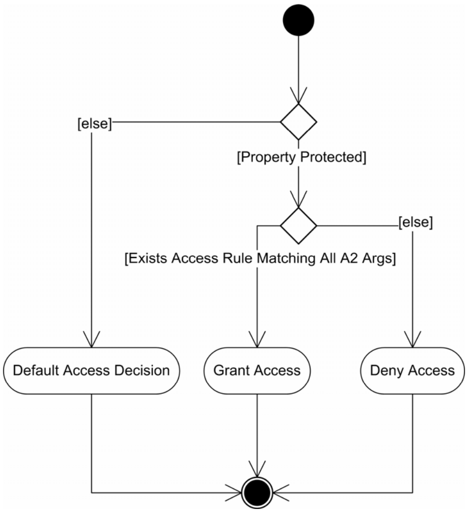

Figure 2: Authorization algorithm diagram.

Graphical schema of the authorization algorithm is presented on figure 2. *Protected property* is a key notion for its understanding. If in the access map exists at least one rule referring to a given device class/property/operation type, we say that the property is protected. If the credentials needed to access a given property with a given operation type are not specified, a *default access decision* is taken. In case the property is not protected, an access rule matching all authorization arguments is searched. If such a rule is found, the operation is allowed. Otherwise, the access is denied.

IMPLEMENTATION DETAILS

As the authorization time is a concern for the CMW operation, it was decided to represent the access map in the CMW server in the form of a *tree*, a separate one for every operation type. Figure 3 presents the structure of access map tree.

Nodes placed with the same distance from the tree root group authorization argument of one type. Authorization is performed by traversing the tree, appropriate in respect to the operation type, from its root to the leaf, trying at each level to match exact authorization parameter or to find a wildcard ‘*’. If this succeeds, the access is granted.

Such a structuring of the access map allows us to assure authorization with the complexity of $O(\log n)$ instructions, where n is the number of access rules. Results of access map performance tests are presented in [1].

Furthermore, the CMW client library was modified to allow transferring RBAC tokens to the CMW server. Finally, callback interfaces were modified to allow reception of RBAC-specific access exceptions.

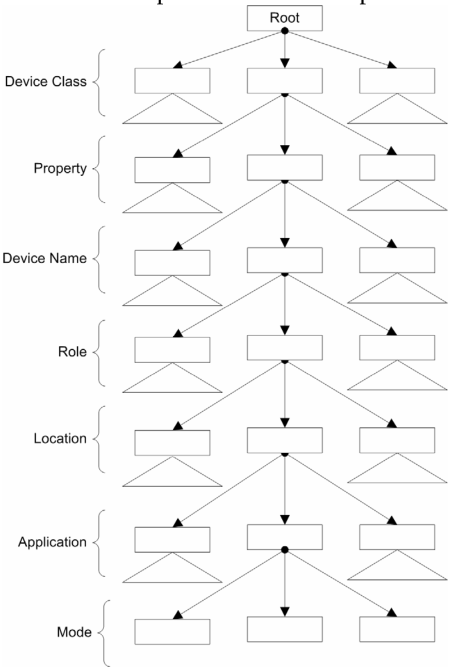

Figure 3: Access map tree structure.

CURRENT STATE

Implementation phase of the RBAC project in the RDA software library is nearly accomplished. The coming months will bring to the operational state the distribution of the accelerator modes in the CMW server.

Newly rebuilt CMW servers already contain RBAC modules. Additionally, equipment specialists are asked to define access maps for their device classes.

Current default access rules grant access for all types of operations. However, transition to the operational state will imply allowing by default get/monitor on operations and denying default access for set operations.

REFERENCES

[1] S. Gysin, C. Schumann, A. Petrov, P. Charrue, V. Kain, K. Kostro, G. Kruk, S. Page, “Role-Based Access Control for the Accelerator Control System at CERN”, ICALEPCS’07, Knoxville, Tennessee, U.S.A.

[2] A. Petrov, C. Schumann, S. Gysin, “User Authentication for Role-Based Access Control”, ICALEPCS’07, Knoxville, Tennessee, U.S.A.

[3] K. Kostro, V. Baggiolini, F. Calderini, F. Chevrier, S. Jensen, R. Swoboda, N. Trofimov, “Controls Middleware – the New Generation”, EPAC’02, Paris, France, p. 2028.

[4] N. Trofimov, V.Baggiolini, S. Jensen, K. Kostro, F. Di Maio, A. Risso, „Remote Device Access in the New CERN Accelerator Controls Middleware”, ICALEPCS’01, San Jose, California, U.S.A., p. 496.