

Chapter 8. Access Control and Authorization

Table of Contents

- Discretionary Access Control
Mandatory Access Control
Role Based Access Control

Access control mechanisms are a necessary and crucial design element to any application's security. In general, a web application should protect front-end and back-end data and system resources by implementing access control restrictions on what users can do, which resources they have access to, and what functions they are allowed to perform on the data.

Authorization and Access Control are terms often mistakenly interchanged. Authorization is the act of checking to see if a user has the proper permission to access a particular file or perform a particular action, assuming that user has successfully authenticated himself.

Access Control refers to the much more general way of controlling access to web resources, including restrictions based on things like the time of day, the IP address of the HTTP client browser, the domain of the HTTP client browser, the type of encryption the HTTP client can support, number of times the user has authenticated that day, the possession of any number of types of hardware/software tokens, or any other derived variables that can be extracted or calculated easily.

Before choosing the access control mechanisms specific to your web application, several preparatory steps can help expedite and clarify the design process;

- 1. Try to quantify the relative value of information to be protected in terms of Confidentiality, Sensitivity, Classification, Privacy, and Integrity related to the organization as well as the individual users. Consider the worst case financial loss that unauthorized disclosure, modification, or denial of service of the information could cause. Designing elaborate and inconvenient access controls around unclassified or non-sensitive data can be counterproductive to the ultimate goal or purpose of the web application.
2. Determine the relative interaction that data owners and creators will have within the web application. Some applications may restrict any and all creation or ownership of data to anyone but the administrative or built-in system users. Are specific roles required to further codify the interactions between different types of users and administrators?
3. Specify the process for granting and revoking user access control rights on the system, whether it be a manual process, automatic upon registration or account creation, or through an administrative front-end tool.
4. Clearly delineate the types of role driven functions the application will support. Try to determine which specific user functions should be built into the web application (logging in, viewing their information, modifying their information, sending a help request, etc.) as well as administrative functions (changing passwords, viewing any users data, performing maintenance on the application, viewing transaction logs, etc.).
5. Try to align your access control mechanisms as closely as possible to your organization's security policy. Many things from the policy can map very well over to the implementation side of access control (acceptable time of day of certain data access, types of users allowed to see certain data or perform certain tasks, etc.). These types of mappings usually work the best with Role Based Access Control.

There are a plethora of accepted access control models in the information security realm. Many of these contain aspects that translate very well into the web application space, while others do not. A successful access control protection mechanism will likely combine aspects of each of the following models and should be applied not only to user management, but code and application integration of certain functions.

Discretionary Access Control

Discretionary Access Control (DAC) is a means of restricting access to information based on the identity of users and/or membership in certain groups. Access decisions are typically based on the authorizations granted to a user based on the credentials he presented at the time of authentication (user name, password, hardware/software token, etc.). In most typical DAC models, the owner of information or any resource is able to change its permissions at his discretion (thus the name). DAC has the drawback of the administrators not being able to centrally manage these permissions on files/information stored on the web server. A DAC access control model often exhibits one or more of the following attributes.

- Data Owners can transfer ownership of information to other users
- Data Owners can determine the type of access given to other users (read, write, copy, etc.)
- Repetitive authorization failures to access the same resource or object generates an alarm and/or restricts the user's access
- Special add-on or plug-in software required to apply to an HTTP client to prevent indiscriminant copying by users ("cutting and pasting" of information)
- Users who do not have access to information should not be able to determine its characteristics (file size, file name, directory path, etc.)
- Access to information is determined based on authorizations to access control lists based on user identifier and group membership.

---

[Prev](#)◆

SSL and TLS◆

[Up](#)

[Home](#)

◆[Next](#)

◆ Mandatory Access Control