

phpGACL

Generic Access Control List

Mike Benoit <ipso@snappymail.ca>
James Russell <james-phpgac1@ps2-pro.com>
Karsten Dambekalns <k.dambekalns@fishfarm.de>

Copyright © 2002,2003, Mike Benoit
Copyright © 2003, James Russell
Copyright © 2003, Karsten Dambekalns

Document Version: 42

Last Updated: **5/20/03 - 18:55:08**

Table of Contents

Advanced usage	21
Using the ACL admin utility	22
API Reference	23
ACL	23
add_acl()	23
edit_acl()	23
del_acl()	24
Groups	24
get_group_id()	24
get_group_parent_id()	24
add_group()	24
get_group_objects()	

edit_object_section()	30
del_object_section()	30
FAQ	31

About

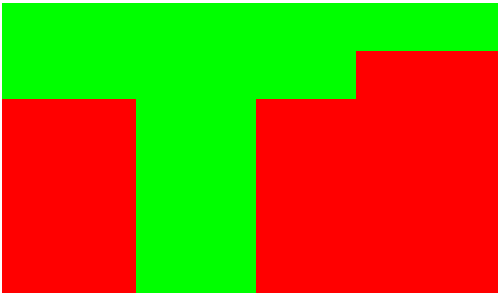
What is it?

phpGACL is a6lt of functions t(Whaallows you to apply access control to arbitrary objects)' 12.6 TL(web pages, da

Introduction

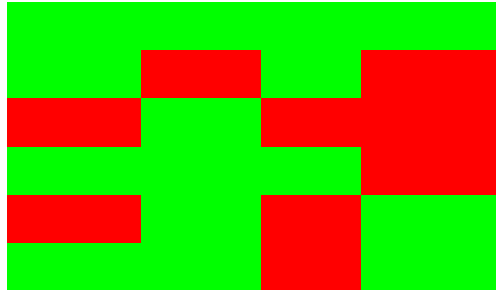
Understanding Access Control

Han is captain of the Millennium Falcon and Chewie is his second officer. They've taken on board some passengers: Luke, Obi-wan, R2D2 and C3PO. Han needs to define access restrictions. Han and Chewie should be able to access the ship's resources, but only if they're not on the ship. Han and Chewie should be able to access the ship's resources, but only if they're not on the ship. Han and Chewie should be able to access the ship's resources, but only if they're not on the ship.



and lengthy adjustment to the matrix, and it's a difficult task to verify that the final matrix is correct.

- It's hard to summarize or visualize. The above example is fairly simple to summarize in a few sentences



Millennium Falcon Passengers

[ALLOW: ALL]

ers

[ALLOW: Lounge]

Another example of fine-grain control happens when the Empire attacks; Han needs to let Luke man the guns, and let R2D2 repair the hyperdrive in the Engine room. He can do this by over-riding the general permissions granted by their status as a "Passenger":

Millennium Falcon Passengers

|—

- Move on to "Passengers", which explicitly says that "Passengers" have Lounge access, so change the internal result to "ALLOW".
- Move to the "Jedi" node, which doesn't mention the Lounge at all.
- Finally move to Luke's node, and again there's nothing there about the Lounge.
- There's nowhere left to go, so the result returned is the current value of the internal result: "ALLOW"

Example 2: We ask: "Does Chewie have access to the Engines?"

- Set the default result, "DENY".
- Work out a path to Chewie:

Millennium Falcon Passengers Crew Chewie

-

This shows how easy it is to granoTw people access. If we used the original matrix scheme,

Han chooses option 3, and removes Chewie from the Engineers list.

Naming Access Objects

phpGACL uniquely identifies each Access Object (AROs, AXOs and ACOs) with a two-keyword combination and its Access Object type.

The tuple "(Access Object type, Section, Value)" uniquely identifies any Access Object.

The first element of the tuple is the type of Access Object (ARO, AXO or ACO).

The second element of the tuple, called the **Section**

Sections are just a way of categorizing Access Objects, to make the user interface more usable, and the code for `acl_check()` more readable. They do not affect the way phpGACL determines access to an object. They cannot be nested (so it would not be able to create a "Males" sub-Section under "Humans" for example; you' d have to create a Section called "Humans-Male" or similar)

Mulss Purposes)

If that' s all you need, that' s fine - AXOs are totally optional.

But because all ACOs are considered equal, it makes it difficult to manage if there are many ACOs. If this is the case, we can change the way we look at Access Objects to manage it more easily.

AXOs are identical to AROs in many respects. There is an AXO tree (separate from the ARO tree), with it' s own Groups and AXOs. When dealing with AXOs, consider an AXO to take the

Keep in mind AXO' s are optional, if you don' t specify an AXO when calling `acl_check()` and a matching ADP exists with no AXO, it will be allowed. However if only APDs exist with AXO' s, and you call `acl_check()` without an AXO, it will fail.

So basically as soon as you specify an AXO when calling `acl_check()`, `acl_check()` will only search ACLs containing AXO' s. If no AXO is specified, only ACLs without AXOs are searched. This in theory (I haven' t benchmarked) gives us a slight performance increase as well.

Advanced setup

Reusing an already existing ADOdb installation

Using phpGACL in your app

Using the ACL admin utility

del_acl()

Deletes permissions already set in the access control list.

del_acl (

inr_acl(o 0 1 tNd1 -23.4d (inReturns: tNd -23.28 Td (inTRUE on suess c, FALSE on failure.j21 13.919 12

)1 1003.28 Tin Ahe accObject()8Tf 2403.28 type_ac"aco", "aro" or "axo")ailur-228f 24Nd1 -23.4d (inRetu

GROUP_IDI(o 0 1 tNd1 -23.4d (inReturns: tNd -23.28 Td (GROUP_PARENT_IDinTRUE on suess

array OBJECT_ID on success, FALSE on failure.

get_object_data()

int TRUE on success, FALSE on failure.

Access Object Sections

The doTut of the API manages the t Sectio that comprisedoTut of the unique name of an Access Obje. t e "t Sectio" for more informaecti.

edit_object_section()

Changes the attributes of a Section. It is not possible to change the Access Object type of a

FAQ