ET: Legacy Development - Feature #932

Tighter interface with Lua admin suite

02.09.2016 12:03 - Spyhawk

Status:	New	**% Done:**	0%
Priority:	Normal	**Spent time:**	0.00 hour
Assignee:			
Category:	Mod generic		
Target version:	2.78		
OS:		**Arch:**	

Description

Some low level code might require a tighter interface with the Lua admin code.
For example, admins shouldn't be kicked by ClientInactivityTimer().

Check the code and add a Lua hook if required.
See G_shrubbot_permission calls in etpub mod code.

Related issues:

Related to ET: Legacy Development - Task # 975: Add WolfAdmin to installers	**Feedback**	**15.01.2017**

Associated revisions

Revision 1c0e3ac7 - 16.03.2017 18:30 - Timothy

Removed obsolete shrubbot checks (refs #932)

History

#1 - 15.01.2017 13:39 - IR4T4

- Related to Task #975: Add WolfAdmin to installers added

#2 - 15.01.2017 13:41 - IR4T4

Some server comands should be moved to Lua ('gib' etc)

#3 - 23.01.2017 11:42 - Timothy

Is it also possible to remove some output in existing lua functions? For example et.MutePlayer also gives feedback to the player that's being muted, which in my opinion doesn't fit in a lua interface, as it should only do what's requested and then let lua handle any output. This also solves the issue of formatting, because at the moment I'm unable to style the output.

#4 - 23.01.2017 12:37 - Spyhawk

Vanilla includes Mute, Unmute, Warn, Kick and Ban. That's the strict minimum for administration.

We should keep these, but move all others commands that have been added to Lua. That includes gib, die, freeze, unfreeze, burn, pip, throw.
RevivePlayer (from vanilla) should probably be moved to Lua too .

#5 - 23.01.2017 16:46 - Dragonji

This is a good idea Spyhawk.

#6 - 23.01.2017 23:23 - IR4T4

When we ship WolfAdmin with installs there is no need for keeping such half implemented commands in qagame.
We need db persistence for bans and mutes anyway and this is given on engine or Lua side. Timo did implement a ban system with ip/ip range bans.
Finally most later/active mods support Lua, WolfAdmin isn't legacy only and the engine can grab sqlite3 data. Let us simply define game manangement is a Lua task.

#7 - 04.02.2017 13:57 - Spyhawk

From Timo:

> !gib is implemented
> !die what's that for?
> freeze,unfreeze cannot since I cannot modify ps.origin
> throw,fling and !launch for the same reason cannot be implemented in Lua only
> burn,pip and !pop also need mod-related edits

See http://dev.timosmit.com/bugtracker/view.php?id=62.

~~TODO: ship relevant pk3 files (such as pm.wav) directly with legacy mod.~~ done

#8 - 16.03.2017 19:29 - Timothy

I took the liberty of removing some obsolete shrubbot calls (see commit:1c0e3ac78752bf3f3a340ae8c3298ec2c28a68ce). I think it is a good idea to decide on the approach we would like to take given the fact that WolfAdmin is able to run standalone and has taken over quite a lot of functionality already. Ideally I would port more of the existing game management functionalities to WolfAdmin, but I would like to leave this up to you. After all, WolfAdmin has to prove itself as a reasonable alternative for what would previously be provided by shrubbot.

At this moment, I am thinking of several routes that we could take. The solutions described in option 1 are a subset of 2, 2 of 3 and so on.

Option 1: bare minimum

By this I intend to get rid of everything related to game management and keep the bare minimum. This excludes any features related to warns, mutes and bans from the mod code, as well as removing the inactivity checks and possibly even forced team balance. After removing this, the game and mod will only provide some mere hooks/lua functions if needed:

- voting hooks (no voting limit, players immune to certain votes);
- chat hooks (team chat spying);
- entity/world modifying functions (e.g. et.setPlayerVelocity for throw,launch)

Option 2: minimum + vanilla features

This option is roughly the same as above, however, in this case inactivity checking and balanced teams can be provided by the mod code to give at least some admin features by default. However, extra hooks may be needed:

- activity hooks (no inactivity putspec or kick);
- team hooks (ignore team balance);

Option 3: basic admin in mod

As the name suggests, also provide minimal admin features, e.g. warn, mute, kick as implemented by vote and/or referee commands. This is kind of the same situation as it is now, which is not ideal for rookie admins as WolfAdmin might provide alternatives and confuse them.

Of course, if any of these options is used, we have to decide on how to implement these hooks/features. Currently et.MutePlayer only sets sess.muted (which can also be set using et.gentity_set) and prints some feedback to the involved player(s), which I find a little cumbersome; a Lua function should only do what is necessary, and leave any output to the calling Lua script. I would also like to hear your opinion on this aspect.

All in all, from my perspective option 1 is the ideal situation: no confusion can arise of duplicate features and a minimum of hooks for a Lua interface is needed. This would entirely separate the player/game management from the mod (except for referees/voting?). I would understand that you find this a rather extreme approach, so any combination of the above is also fine with me. I would just like to know the desired route so I can take this into account for the next iterations of WolfAdmin.

#9 - 29.06.2017 20:20 - IR4T4

- Target version changed from 2.76 to 2.78

#10 - 06.01.2019 09:29 - IR4T4

- Target version changed from 2.78 to 2.77

#11 - 14.02.2019 11:41 - Spyhawk

- Target version changed from 2.77 to 2.78