For non-victorious combat, I'm quite concerned that it assigns veteran levels. In theory it sounds good: you became more experienced in combat and survived.

But the ability to know that certain units in certain situations would never complete combat with a victor... This allows an exploit.

In those known situations, undeclared allies can attack each other's units to promote them.

Suggestion 0: Hard-coded disallowing of promotion is quick and easy but I think this would be mean and disallow a lot of legitimate promotions.

Suggestion 1: server setting promote_without_victory="enabled" or "disabled". This is a LOT easier than Suggestion 2 below and can be put up right away while Suggestion 2 gets procrastinated.

Suggestion 2 - promote_without_victory can be a flag going into units and/or unitclasses. After careful thinking, the number of combinations and situations a ruleset designer has to patch up with these flags can get nasty, and I hope we fix it in a way that does not have gaping holes out of the designer's control, but rather give them full control.

It may be complex but one way for full control is promote_defender_without_victory and promote__without_victory both going in the unit's flags/settings. Having promote_defender_without_victory going into the attacker unit would be important. Why? Many units would be used to strafe or soften targets, but the incentive for such an action is completely wasted if those units get promoted. Plus, designers might come up with who knows what kinds of ideas, such as mustard gas or countless other things meant to weaken but not resulting in defender promotions.

Certain attacking unit types will be the ones that most often create known exploit situations. If UnitClass flags are used, the UnitType flags would override them.

For suggestion 2, instead of 0 or 1 on/off, it could be a percentile. 0 is off, 100 is on, but other numbers would modify the chance of becoming veteran. So for example, an air unit doing strafing run could have promote_defender_without_victory = 50, which would make the chance of promotion for a surviving defender HALF of what it would normally be.

The plan for S3.0 is to live with just already existing only_killing_makes_veteran setting.
We have a ticket of introducing unit experience somewhere around that might fix a lot in promotion mechanics. (But Sławomir’s suggestion might be noteworthy as well, you don’t move, you grow fat on sides.) Maybe before it we could use main veteran_raise_chance scaled to the probability the combat has a victor and veteran_work_raise_chance for the probability both sides survive.

#6 - 2021-10-03 12:48 AM - Marko Lindqvist
Anybody have any concrete implementation ideas - or even code - for this to use in 3.1? We should get this forward relatively soon from blocking S3_1 d3f.

#7 - 2021-10-03 10:36 PM - David Fernandez (bard)
Marko Lindqvist wrote:

The plan for S3_0 is to live with just already existing only_killing_makes_veteran setting.

I have been testing 3.0 with limited combat rounds, and with only_killing_makes_veteran enabled, and to me it seems to be enough to prevent exploits. If I didn’t miss anything, the only way to get veterancy in this case is by killing some unit, so not possible to train allied units without losses.

However, I was missing a way to prevent promotions when the defending unit has defense 0 (or no way to damage the attacker). I hope this ticket can allow something like that too.

#8 - 2021-10-09 05:36 PM - Marko Lindqvist
David Fernandez (bard) wrote:

However, I was missing a way to prevent promotions when the defending unit has defense 0 (or no way to damage the attacker).

--> https://osdn.net/projects/freeciv/ticket/43009

#9 - 2022-01-12 06:34 AM - Marko Lindqvist
I think also the recently introduced combat_odds_scaled_veterancy helps a bit with this. I’m inclined to say the situation is good enough for 3.1 now. We may postpone this ticket to 3.2 to re-evaluate the situation.

#10 - 2022-01-13 08:38 AM - Lexxie L
Marko Lindqvist wrote:

I think also the recently introduced combat_odds_scaled_veterancy helps a bit with this. I’m inclined to say the situation is good enough for 3.1 now. We may postpone this ticket to 3.2 to re-evaluate the situation.

Yeah OK. The possibility of the exploit seemed scary but the real world cases just don’t seem to happen -- at least in our rulesets.

#11 - 2022-03-12 09:32 AM - Marko Lindqvist
- Sprint/Milestone changed from 3.1.0 to 3.2.0

Marko Lindqvist wrote:

We may postpone this ticket to 3.2 to re-evaluate the situation.

Doing just that.

#12 - 2022-03-12 09:33 AM - Marko Lindqvist
- Blocks Task #939772: S3_2 datafile format freeze (d3f) added

#13 - 2022-03-12 09:33 AM - Marko Lindqvist
- Blocks deleted (Task #673656: S3_1 datafile format freeze (d3f))