redmineorg-copy202205 - Vote #78018

Use more secure hashing algorigthm

2022/05/09 18:08 - Admin Redmine

ステータス:	New	**開始日:**	2022/05/09
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	Accounts / authentication_7	**予定工数:**	0.00時間
対象バージョン:		**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/24520	**status_id:**	1
category_id:	7	**tracker_id:**	2
version_id:	0	**plus1:**	0
issue_org_id:	24520	**affected_version:**	
author_id:	163201	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	1		

説明

h2. Introduction

Currently the hashing algorithm used is: @SHA1@ [0].
I suggest to use a more secure (computationally expensive) algorithm to store the password. Some alternative algorithms to use:

- @bcrypt@ with reasonable iteration count.
- @scrypt@.

h2. Drawbacks

The only drawback I can think of is the migration of the database to use the new algorithm. I'm thinking about using this approach to fix this issue:

Let's call the new secure hashing algorithm: @H@.

- The salt will be kept in the database.
- Foreach user in the database, *replace* the hashed password: @SHA1($salt.$plain_password)@ with @H(SHA1($salt.$plain_password)@.
- The algorithm @H(SHA1($salt.$plain_password)@ will be used from now when creating a new users/resetting a new password ...

h2. Why is @SHA1@ insecure ?

When I say insecure I'm not talking about the collision ratio. I'm referencing that it's easy (fast) to compute.
Example: Using hashcat[1] v3.10 with GPU: R9 290X (+10Mhz) - AMDGPU-pro 16.40[2], It's able to compute:

- 4,102,360,845 @sha1@ hash per second.
- 94,960 @scrypt@ hash per second.
- 12,070 @bcrypt@ hash per second (cost of 10 iirc).

Thoughts ?

[0] https://github.com/redmine/redmine/blob/master/app/models/user.rb#L840
[1] https://hashcat.net/
[2] https://docs.google.com/spreadsheets/d/1B1S_t1Z0KsqByH3pNkYUM-RCFMu860nlfSsYEqOoqco/edit#gid=1591672380

journals

related_issues

relates,New,36056,Update password hash function

履歴

#1 - 2022/05/10 17:06 - Admin Redmine

- カテゴリ を Accounts / authentication_7 にセット