OTPme - Feature #70

Implement argon2 support for AES key derivation

31 December 2015 20:08 - The 2nd

Status:	In Bearbeitung	**Start date:**	31 December 2015
Priority:	Normal	**Due date:**	
Assignee:	The 2nd	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	OTPme 0.3		

Description

- https://password-hashing.net/
- https://pypi.python.org/pypi/argon2

History

#1 - 01 January 2016 19:49 - The 2nd

- Maybe we should use pyzxcvbn to implement dynamic iterations based on password strength
- We also should consider dynamic calculation of aron2's "m" parameter based on available host memory

#2 - 03 January 2016 20:52 - The 2nd

- The python version of zxcvbn has some issues -> https://github.com/dropbox/python-zxcvbn/issues/14
- And the original javascript version fails to generate a reasonable score on passwords with very common patterns

```
zxcvbn('SuziMiller_2016');
{ password: 'SuziMiller_2016',
  guesses: 188140000000,
  guesses_log10: 11.274481139688914,
  sequence:
   [ { pattern: 'dictionary',
       i: 0,
       j: 3,
       token: 'Suzi',
       matched_word: 'suzi',
       rank: 3134,
       dictionary_name: 'female_names',
       reversed: false,
       l33t: false,
       base_guesses: 3134,
       uppercase_variations: 2,
       l33t_variations: 1,
       guesses: 6268,
       guesses_log10: 3.797128987796552 },
     { pattern: 'dictionary',
       i: 4,
       j: 9,
       token: 'Miller',
       matched_word: 'miller',
       rank: 7,
       dictionary_name: 'surnames',
       reversed: false,
       l33t: false,
       base_guesses: 7,
       uppercase_variations: 2,
       l33t_variations: 1,
       guesses: 50,
       guesses_log10: 1.6989700043360185 },
     { pattern: 'bruteforce',
       token: '_2016',
       i: 10,
       j: 14,
```

```
      guesses: 100000,
      guesses_log10: 5 } ],
  calc_time: 7,
  crack_times_seconds:
   { online_throttling_100_per_hour: 6773040000000,
     online_no_throttling_10_per_second: 1881400000,
     offline_slow_hashing_1e4_per_second: 18814000,
     offline_fast_hashing_1e10_per_second: 18.814 },
  crack_times_display:
   { online_throttling_100_per_hour: 'centuries',
     online_no_throttling_10_per_second: '59 years',
     offline_slow_hashing_1e4_per_second: '7 months',
     offline_fast_hashing_1e10_per_second: '19 seconds' },
  score: 4,
  feedback: { warning: '', suggestions: [] } }
```

- There is another python password strength check but it also fails on the same password -> https://github.com/cadithealth/passwordmeter

```
# python
Python 2.7.9 (default, Apr  2 2015, 15:33:21)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import passwordmeter
>>> password = "SuziMiller_2016"
>>> strength, improvements = passwordmeter.test(password)
>>> print strength
0.915112131644
```

#3 - 06 August 2018 20:24 - The 2nd

http://www.1pw.de/brute-force.html

#4 - 07 August 2018 14:05 - The 2nd

- Status changed from Neu to In Bearbeitung