NAME

apgbfm − APG Bloom filter management program

SYNOPSIS

apgbfm -f *filter* **-n numofwords** [**-q**] [**-s**]
apgbfm -f *filter* **-d** *dictfile* [**-q**] [**-s**]
apgbfm -f *filter* **-a word** [**-q**]
apgbfm -f *filter* **-A** *dictfile* [**-q**]
apgbfm -f *filter* **-c word** [**-q**]
apgbfm -f *filter* **-C** *dictfile* [**-q**]
apgbfm -i *filter*
apgbfm [**-v**] [**-h**]

DESCRIPTION

apgbfm is used to manage Bloom filter that is used to restrict password generation in **APG** pasword generation software. Usage of the Bloom filter allows to speed up password check for large dictionaries and has some other benefits.

The idea to use Bloom filter for that purpose is came from the description of the **OPUS** project **OPUS: Preventing Weak Password Choices** *Purdue Technical Report CSD-TR 92-028* writen by *Eugene H. Spafford*.

You can obtain this article from:
http://www.cerias.purdue.edu/homes/spaf/tech-reps/9128.ps
It has very nice description of Bloom filter and it's advantages for password checking systems.

In simple words, **apgbfm** generates *n* hash values for every word and sets corresponding bits in filter file to 1. To check the word **apgbfm** generates the same hash functions for that word and if all *n* corresponding bits in filter file are set to 1 then it suppose that word exists in dicionary. **apgbfm** uses **SHA-1** as a hash function.

apgbfm can be used as standalone utility, not only with **apg**, or **apgd**.

WARNING !!!
Filter file format can be changed in the future. I'll try to make file formats compatible but i can not guaranty this.

WARNING !!!
apgbfm may slow down your computer during filter creation.

OPTIONS

-f *filter* use *filter* as the name for Bloom filter filename.

-i *filter* print information about *filter*.

-n numofwords
create new empty filter for **numofwords** number of words. Useful when you want to fill filter dynamicaly.

-d *dictfile*
create new filter from *dictfile*. It may take a lot of time to generate filter from a big dictionary. In that dictionary you may place words (one per line) that should not appear as generated passwords. For example: user names common words, etc. You even can use one of the dictionaries that come with *dictionary password crackers*. This check is case sensitive. For example, if you want to reject word 'root', you should insert in *dictfile* words: root, Root, RoOt, ... , ROOT. To indicate that program is working **apgbfm** prints dot for every 100 words added in dictionary.

-a word
add **word** to the filter.

-A *dictfile*
add all words from *dictfile* to the filter. To indicate that program is working **apgbfm** prints dot for every 100 words added in dictionary.

-c word
check **word** for appearance in the filter.

-C *dictfile*
check every word from *dictfile* for appearance in the filter.

-q quiet mode.

-s create new filter in case-insensitive mode.

-v print version information.

-h print help information.

EXIT CODE

On successful completion of its task, **apgbfm** will complete with exit code 0. An exit code of -1 indicates an error occurred. Textual errors are written to the standard error stream.

FILES

None.

BUGS

None. If you've found one, please send bug description to the author.

This man page is Alpha too.

SEE ALSO

apg(1)

AUTHOR

Adel I. Mirzazhanov, <a-del@iname.com>
Project home page: http://www.adel.nursat.kz/apg/