AAPR : AES Archive Password Recovery ver 0.01

Introduction

AAPR will aim at being a password recovery utility for RAR and ZIP files encrypted using the AES encryption scheme. However, in its first release, it only support AES encrypted RAR (Read RAR v3+), that have encrypted file names. If you try to open a RAR file in Winrar and it does not ask you straightaway for a password and you can see the names of the encrypted files, this utility cannot help you, yet.

Because there is no known flaws in AES, the only way to go about finding out a password is to try until you find the right one and with AES it can takes a long times.

AAPR ver 0.01 is a single threaded application and depending on the speed of your processor, should be able to test between 10 and 20 passwords a seconds, on a modern computer : if the password is not weak, it's going to take a long long time.

AAPR should works on most POSIX compliant OS (Unix, Linux, Mac OS X) and windows. Binaries are provided for Windows and Mac OSX. You can compile the utility yourself just by typing 'make'.

The brute-force Attack

You can try to find a password by trying all the passwords combinations made up of a specific character set. If you don't know anything about the password, that would likely be at least the alphabetical lower-case characters and the numbers.

To do that you must first, create a text file that contains the characters you want to include in you character set :

open a text editor.
write all the characters you want on the first line.
save it.

ex :

charset.txt :
abcdefghijklmnopqrstuvwxyz1234567890

Then launch aapr with the following command :
aapr -mb <charset filename> -p 7 <encryted rar filename>

-p 7 means that AAPR will try all passwords up to 7 characters in length.

The brute-force attack method allows you to form password with character string instead of single characters. This is useful when you suspect that a password is combined of known elements but don't know which.

To use that feature, just create a file containing each elements on a new line :

ex :

string_set.txt:

word1
word2
word3
...

Type the same command : ***aapr -mb <charset filename> -p 7 <encryted rar filename>***
Here -p7 means that it will try all the passwords formed with the concatenation of the words in the <charset_filename> up to 7 elements.

The dictionary attack :

With this simple method, you need to use a text file containing a list of words (one per line), and the software will try each word as a potential password until the list ends.

This is a useful method to try to find weak password in a limited time : you should be able to test half-a-million words in about 12 hours.

To launch such an attack : ***aapr -md <dictionary filename> <encryted rar filename>***

Other Features :

Automatic Progression Saving :

By default AAPR will save its progression every 1500 tests by writing to a .crk file with the same name as the archive. This allows you to stop AAPR at any time and be able to start it again later.

To restart a previously started job type : aapr -c <encrypted rar filename>

You can specify your own saving interval with the -t options.
ex : ***aapr -md <dictionary filename> -t 2000 <encryted rar filename>***
This will save every 2000 tries instead of the 1500 default.

Warning : if you want to change the settings on a previously launched decryption attempt, you need to delete de .crk file associated with the archive you are looking to decrypt. Otherwise, it will just continue the previous attack.

Benchmark :

If you want an idea of how long a particular job will take, you can use the benchmark feature. Type all the options like you would to start the actual job but add the options -b followed by a limit of password to try.

ex : ***aapr -md <dictionary filename> -b 1000 <encryted rar filename>***

AAPR will then stop after 1000 try and tell you how fast things are going.

Note : if you specified more than the saving limit, don't forget to delete the .crk file to start the real calculation.

Index Range :

You can start at any point in the password space you are trying.
If you are trying to find a password with a character set of 10 characters and with a maximum length of 5, there are 66429 possible password.

You can tell AAPR to start looking a the #-th password and stopped at the #-th password in the list of possible passwords using the -i option.

ex: ***aapr -md <dictionary filename> -i 3000 5000 <encryted rar filename>***
AAPR will only try passwords between the 3000th and 5000th words in the dictionary list.

This feature is useful if you want to split the work load on several computers to speed things up. You could also use it to launch 2 instances of AAPR on the same machine in case you have a multi-core/processor machine as AAPR is at present single threaded. Finally this is also how the resuming from previously started job works : AAPR save the current index in the .crk file with the others settings.

Note : you can find out the number of possible passwords with you chosen settings using the benchmark feature to find out how to split the work.

Note : when launching several instance of the application, note that you don't need to copy the full RAR file several times : the utility only uses the first few thousands byte of the Rar files so you can use a hexadecimal editors to copy just the beginning of the file instead of having multiple copies.

Test function

Finally there is a hidden testing function to check that the utility is working properly on your computer : type aapr -?
If an error message comes up, please send me a mail at glachesis at users.sourceforge.net

Remember that any passwords that is long enough and not present in a dictionary will not be found in a reasonable time with this utility and there is nothing that can be done.