

Forum:[Hack Forums](#)

Thread: [Advanced options for PRINCE mode on JtR](#)

permanoob - January 10, 2017, 7:55 pm

****props to chick3nman for his help in another thread**

I'm trying to use PRINCE mode on JtR to combine words from a manual word list to find passphrase for a PGP Desktop .skr. Because I know that the pass is a keyboard walk, I know that the "words" used to compile the lists won't be repeated. Also, there are multiple walks from NW to SE on one side of the keyboard (US Eng Qwerty) and symmetric walks from NE to SW on the other side of the keyboard--such as 1qaz2wsx=[;.-pl,. Because of this, I've built wordlists for 4 char walks, 3 char walks and 2 char walks to eliminate non-symmetric candidates with PRINCE. Finally, I know the walks happen from outside in (i.e. 1qaz before 2wsx).

Questions:

- 1) How do I tell PRINCE in JtR to not repeat a word from the wordlist (no 1qaz1qaz....)?
- 2) How do I prioritize one walk over another (so left side such as 1qaz) always comes first? This would help even if it must be manually concatenated (1qaz plus PRINCE of remaining keyspace).
- 3) Are any additional parameters required for PGP Desktop Disk in JtR 1.8.0 Jumbo other than processing .skr with gpg2john?

Thanks in advance.

FWIW, I went through 4 char walk wordlist at ~4000c/s and completed in 1:33:00 for min-length=16 and max-length=24. Without optimizing JtR/PRINCE, 3 char walk wordlist with min-length=12 and max-length=24 will take until the year 2037. :-/

chick3nman - January 10, 2017, 8:53 pm

You processed the skr with gpg2john so you have the formatted "hash" for JTR correct? Could you post that or send that over for testing so that I can get an idea of how speeds and times stack up with different attacks? I might be able to come up with some better ways of doing it if I can try them out and see how well they perform.

permanoob - January 10, 2017, 9:22 pm

Thanks, Chick3nman. Sent PM.

chick3nman - January 11, 2017, 11:29 am

(Jan 10, 2017 10:22 PM)permanoob Wrote: [CITE](#) Thanks, Chick3nman. Sent PM.
Just started doing some tests. Wordlists tests are going well, good speeds right now.

25597p/s 25597c/s 25597C/s -'=%#..-!>*

What is your command setup right now? I'm building the lists before and using them in john for better duplicate removal but I might move to using kwproc or the prince processor and straight piping it.

permanoob - January 11, 2017, 2:11 pm

I'm currently using the following command line:

```
john /root/Desktop/PGP/gpg2johnskfile.lst --prince=/root/Desktop/wordlist3.txt --min-length=12  
--max-length=18
```

Due to resource constraints at my current location, this is a standalone Kali box with no discrete GPU. :/

Currently getting 0g 0:16:11:39 11.02% (ETA: 2017-01-17 03:50) 0g/s 4233p/s 4233c/s 4233C/s
PL<4rfuhbOKM2ws4rf

Wondering if piping from PRINCE might help? Pruning input candidates certainly will but my questions on that above remain.

Thanks for your continued effort.

chick3nman - January 11, 2017, 2:32 pm

Using the princeprocessor that comes with hashcat-utils(i think its included, might be separate) may be a useful way to separate that part of the generation. I'm not sure how to avoid the duplicates you mentioned unless you are dumping into a file first and then using that file in john after cleaning it. Also I'm not using my GPUs for this, not sure john supports GPU for this algorithm. Just one big CPU :)

permanoob - January 12, 2017, 7:28 am

Still working to clean the input files and better sort, but the PRINCE function of JtR doesn't seem to have many configurable options. I'm trying to figure out the hashcat one now.

chick3nman - January 12, 2017, 5:24 pm

(Jan 12, 2017 08:28 AM)permanoob Wrote: [CITE](#) Still working to clean the input files and better sort, but the PRINCE function of JtR doesn't seem to have many configurable options. I'm trying to figure out the hashcat one now.

Both the prineprocessor and kwprocessor may be useful here though I'm still not sure whats best since I'm not certain i understand what your exact possible keyboard walk looked like. Starting on the left side but then moving diagonal?

permanoob - January 13, 2017, 1:28 pm

The walk started with the left side moving top left (NW) to bottom right (SE) for multiple diagonal columns from the left to the right. There then was an equivalent number of diagonal columns/keystrokes on the right side with a NE to SW walk. Right side most likely is shifted.

Example: 1qaz 3edc +{:>)OKM (spaces added for clarity here but no spaces in passphrase)

chick3nman - January 13, 2017, 6:02 pm

I see. So looks like its 2 "blocks" on each side? Or rather, 2 columns? I didn't put a whole lot into cracking the hash itself, just benchmarked it and ran some tests using the prince mode to see if i understood its function. I can dedicate a good bit of attack time this weekend to see if i can recover it for you.

permanoob - January 16, 2017, 7:04 am

Was off the grid for the weekend. Sorry I missed your message! It was at least two blocks and no more than three on each side with equal numbers per side (as far as I recall).

I let mine run for quite a while but haven't had a hit yet.

chick3nman - January 16, 2017, 2:34 pm

(Jan 16, 2017 08:04 AM)permanoob Wrote: [\[CITE\]](#) Was off the grid for the weekend. Sorry I missed your message! It was at least two blocks and no more than three on each side with equal numbers per side (as far as I recall).

I let mine run for quite a while but haven't had a hit yet.

I put it through a little bit but didn't do all that much yet. I'm working on generating a list of possible stems or short walks to build off of.