

Forum:[Hack Forums](#)

Thread: [Split passwords from a file](#)

I3k0w - June 18, 2016, 9:15 am

I tried using crunch and hydra for generating passwords and brute-forcing. But hydra's limit is 50,000,000 lines per file and my file was exceeding that limit. So I thought it will be a good idea if I try and create program that splits the file into separate files.

So I came up with this. I know it's not something complicated, but maybe it can help someone ?

Code def main():

```
fr_name = input("Enter the name of the file to read from: ")
num_passwords = int(input("Enter the number of passwords per file: "))
```

```
n = 0
k = 1
```

```
fw = open("1.lst", "w")
```

```
with open(fr_name) as fr:
```

```
    for line in fr:
```

```
        if(n == num_passwords):
```

```
            n = 0
```

```
            k = k + 1
```

```
        if(not fw.closed):
```

```
            fw.close()
```

```
        fw = open(str(k) + ".lst", "w")
```

```
    #end
```

```
    fw.write(line)
```

```
    n = n + 1
```

```
    #end
```

```
#end
```

```
fw.close()
```

```
print("The passwords were split into " + str(k) + " separate files.")
```

```
#end
```

```
if __name__ == "__main__":
```

```
    main()
```

Q: What are you guys using for splitting a file/generating passwords? Does crunch has a command for splitting?

C0ldplay - June 18, 2016, 9:18 am

Easier way...

dd if=crunch.one of=Crunch.one.partone bs=1M count=50 skip=0 creates a file named

Crunch.one.partone with 50 but that can be changed to your liking.

I3k0w - June 18, 2016, 10:02 am

(Jun 18, 2016 09:18 AM)rob_ert Wrote: [\[CITE\]](#) Easier way...

dd if=crunch.one of=Crunch.one.partone bs=1M count=50 skip=0 creates a file named Crunch.one.partone with 50 but that can be changed to your liking.
Nice one!

But can you explain what bs, count and skip are for and how we can use them?

C0ldplay - June 19, 2016, 1:15 am

(Jun 18, 2016 10:02 AM)I3k0w Wrote: [\[CITE\]](#) Nice one!

But can you explain what bs, count and skip are for and how we can use them?
bs= sets the block size. For example, bs=1 would set it to 1MiB.
count= copies the set amount of blocks.

Goodies - June 19, 2016, 7:29 am

Lol don't use dd. dd uses block sizes, not line counts. This is statistically very likely to end up somewhere in the middle of a word/sentence...

There's an actual split utility in linux:
<http://linux.die.net/man/1/split>

This will split a line into multiple files with no more than 10,000,000 lines each:
Code split -l 10000000
The default naming is pretty strange. Use -d for part numbers instead of letters (and -S if you think there are going to be more than 100 files. In your case, I doubt it.

If you actually want a Python version, though. You can do something like this:

```
PHP Code
<?php
import os.path

count = 0
filename = "test.txt"
max_lines = 1
fname, ext = os.path.splitext(os.path.basename(filename))
template = "{}.{:03}{}"
fout = None
with open(filename) as fin:
    for i, line in enumerate(fin):
        if not i % max_lines:
            if fout:
                fout.close()
            fout = open(template.format(fname, count, ext), "w")
            count += 1
        fout.write(line)
```