Julien Marchand • Dev blog

Yet another dev blog

Using the Mega API: how to upload a file anonymously (without logging in).

I received some emails asking me how to upload a file anonymously using the API, since it is possible to upload a file from the Mega.co.nz website without logging in. I actually gave a solution in the comments of this article, but it’s not very easy to find, so I’m posting it here 😀

“Anonymous” uploads use ephemeral accounts, as described in the developer’s guide:

> *MEGA supports ephemeral accounts to enable pre-registration file manager operations, allowing applications to present MEGA’s functionality with a lowered barrier to entry. They naturally do not have an e-mail address associated with them, the password (master key) is generated randomly, and they are subject to frequent purging. In a browser context, the credentials for ephemeral accounts live in the DOM storage.*

Thus, we actually have to log in, but with a freshly generated ephemeral account. Let’s see how it works:

```
def login_anon():
  global sid, master_key
  master_key = [random.randint(0, 0xFFFFFFFF)] * 4
  password_key = [random.randint(0, 0xFFFFFFFF)] * 4
  session_self_challenge = [random.randint(0, 0xFFFFFFFF)] * 4

  user_handle = api_req({
      'a': 'up',
      'k': a32_to_base64(encrypt_key(master_key, password_key)),
      'ts': base64urlencode(a32_to_str(session_self_challenge) + a32_to_str(encrypt_key(session_self_challenge, master_key)))
  })

  print "ephemeral user handle: %s" % user_handle
  res = api_req({'a': 'us', 'user': user_handle})

  enc_master_key = base64_to_a32(res['k'])
  master_key = decrypt_key(enc_master_key, password_key)
  if 'tsid' in res:
    tsid = base64urldecode(res['tsid'])
    if a32_to_str(encrypt_key(str_to_a32(tsid[:16]), master_key)) == tsid[-16:]:
      sid = res['tsid']
```

We randomly generate a master key, a “password key” (equivalent to the hash of a regular user’s password) to encrypt to master key, and a session self challenge (that will be used to check the generated password and get the session ID, since our ephemeral account does not have a RSA key pair).

Then, the *getfiles()* and *uploadfile()* functions are the same as in my first article. So, let’s upload a file anonymously and get its public URL to share it on the web:

```
login_anon()
getfiles()
uploaded_file = uploadfile('/home/julienm/mega/test_file.png')
print getpublicurl(uploaded_file['f'][0])
```

We have to call the *getfiles()* function to get the ID of the root node, to which we are uploading our file. The *uploadfile()* method is simply modified to change the final “print” into a “return” and return informations about the uploaded file.

The *getpublicurl()* method gets the public handle of the file (that is not the same as the “private” handle that you see in the ‘*h*‘ attribute when listing your files, you can actually enable or disable the public handle for a file, whether you want it to be public or not), decrypts its key, and concatenates the two informations to obtain the public URL:

```
def getpublicurl(file):
  public_handle = api_req({'a': 'l', 'n': file['h']})
  key = file['k'][file['k'].index(':') + 1:]
  decrypted_key = a32_to_base64(decrypt_key(base64_to_a32(key), master_key))
  return "http://mega.co.nz/#!%s!%s" % (public_handle, decrypted_key)
```

That’s it! It works 🙂

```
julienm@rchand:~/mega$ python anon_upload.py
ephemeral user handle: 5hq-EIBu_yc

http://mega.co.nz/#!V51SVYzY!pMS4P8hyBqFBC3Qdh0YNG4xEbJ8Kj8dYQFuxdDt6dMU

julienm@rchand:~/mega$
```

I'm going to post part 3 of the MegaFS series very soon, as well as a PHP and a Java version of all my examples. Stay tuned

Google+

Share this: Like 0 Tweet Digg [redacted] More

This entry was posted in Uncategorized on February 6, 2013 [/web/20140402040831/http://julien-marchand.fr/blog/using-the-mega-api-how-to-upload-a-file-anonymously-without-logging-in/] .

7 thoughts on "Using the Mega API: how to upload a file anonymously (without logging in)."

DinoEO
February 6, 2013 at 10:59 pm

Salut,

On pourrait avoir la version PHP aussi SVP

Kentin

February 8, 2013 at 12:44 pm

Merci du partage Julien !
Même demande que DinoEO 🙂
Je n'ai pas encore comprit tout le process de décryptage du fichier,
est-il identique à celui du nom du fichier ? dec_attr() ?

Neozaru

February 10, 2013 at 8:08 pm

Non Kentin, le déchiffrage est différent pour les fichiers.
Jusqu'ici, c'était de l'AES CBC qui était utilisé (pour faire court, chaque "bloc" du message à chiffrer était chiffré en fonction du bloc précédent).
Dans le cas d'un transfert de fichier, c'est de l'AES CTR qui est utilisé (chaque "bloc" du message à chiffrer est chiffré en fonction d'un compteur et ne dépend pas d'un calcul précédent, ce qui permet de télécharger/déchiffrer un fichier en utilisant plusieurs threads)
Si ça t'intéresse, c'est bien expliqué sur Wikipédia :
http://fr.wikipedia.org/wiki/Mode_d'op%C3%A9ration_(cryptographie)#Encha.C3.AEnement_des_blocs_:_.C2.AB_Cipher_Block_Chaining_.C2.BB_.28CBC.29

En regardant vite fait ton code PHP, et d'après cette page (http://php.net/manual/fr/mcrypt.constants.php), il ne semble pas d'y avoir de support de l'AES CTR pour MCrypt (mais je me trompe peut-être).
A priori, il y a "phpseclib" qui fait du bon boulot (exemples d'utilisation ici : http://phpseclib.sourceforge.net/crypt/examples.html). C'est normalement plus performant que le "mcrypt" de base, et ça supporte l'AES CTR. Tu peux tenter

Julien, as-tu compris pourquoi il faut chiffrer chaque portion du MAC avant de le comparer ? N’y a-t-il pas moyen de comparer les versions “en clair” ? (je ne vois pas à quel niveau cela constitue une sécurité supplémentaire).

Enfin, j’oublie l’object principal de mon commentaire : MERCI pour ces articles. Cela permet de bien piger comment fonctionne Mega, et d’en apprendre plus sur les algos de chiffrement. (cerise sur le gâteau, j’ai retiré “Python” de mon CV en regardant ton code)

Bonne continuation

:)
February 12, 2013 at 5:19 pm

I’ll wait for php or Java IDE.. if u can get it u’re my new heroe

:)
February 12, 2013 at 5:30 pm

Imposible to get it…

In windows it fails every time and with linux (Ubuntu) I can’t…

```
Traceback (most recent call last):
File “test.py”, line 300, in
getfile(‘RtQFAZZQ’, ‘OH8OnHm0VFw-9IzkYQa7VUdsjMp1G7hucXEk7QIZWvE’)
File “test.py”, line 264, in getfile
```

dl_url = file['g']
TypeError: ‘int’ object has no attribute ‘__getitem__’

Do you know what hapens?

FLM
March 22, 2013 at 5:05 pm

in api_req there is url = ‘https://g.api.mega.co.nz/cs?id=%d%s’ % (seqno, ‘&sid=%s’ % sid if sid else ”)

change the ” & ” to “&”
url = ‘https://g.api.mega.co.nz/cs?id=%d%s’ % (seqno, ‘&sid=%s’ % sid if sid else ”)

it’s an html character

FLM
March 23, 2013 at 7:52 pm

When uploading a large file ex: (700MB) it only uploads 143MB

it is due to anonymous upload or it’s a mega bug ??