Combo - Bug #77545

chargement d'une ressource (DecompressionBombError: Image size (200873929 pixels) exceeds limit of 178956970 pixels, could be decompression bomb DOS ...)

12 mai 2023 15:57 - Sentry Io

Statut:	Nouveau	**Début:**	12 mai 2023
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Chargement d'une ressource, je ne sais pas quelle taille de fichier mais vraiment trop gros.

(attraper l'erreur et dire que vraiment trop gros)

Plus loin, #61703.

https://sentry.entrouvert.org/entrouvert/publik/issues/110313/

```
DecompressionBombError: Image size (200873929 pixels) exceeds limit of 178956970 pixels, could be
decompression bomb DOS attack.
  File "sorl/thumbnail/base.py", line 104, in get_thumbnail
    source_image = default.engine.get_image(source)
  File "sorl/thumbnail/engines/pil_engine.py", line 73, in get_image
    return Image.open(buffer)
  File "PIL/Image.py", line 2944, in open
    im = _open_core(fp, filename, prefix, formats)
  File "PIL/Image.py", line 2931, in _open_core
    _decompression_bomb_check(im.size)
  File "PIL/Image.py", line 2841, in _decompression_bomb_check
    raise DecompressionBombError(

Image size (200873929 pixels) exceeds limit of 178956970 pixels, could be decompression bomb DOS a
ttack.
```

Demandes liées:

Lié à Combo - Development #61703: redimensionner automatiquement les images t...	**Nouveau**	**11 février 2022**
Lié à Combo - Development #86995: vérifier que les images transférées en ress...	**Solution déployée**	**15 février 2024**

Historique

#1 - 12 mai 2023 15:57 - Frédéric Péters

- Projet changé de Suivi des traces à Combo

#2 - 12 mai 2023 15:57 - Frédéric Péters

- Lié à Development #61703: redimensionner automatiquement les images transférées en ressources ajouté

#3 - 10 janvier 2024 17:30 - Yann Weber

Je n'arrive pas a reproduire exactement la même erreur. Mais j'arrive quand même a faire lever une DecompressionBombError :

```
Internal Server Error: /manage/assets/upload/
Traceback (most recent call last):
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/core/handlers/exception.py", line 4
7, in inner
    response = get_response(request)
               ^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/core/handlers/base.py", line 181, i
n _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/contrib/auth/decorators.py", line 21, in _wrapped_view
    return view_func(request, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/views/generic/base.py", line 70, in view
    return self.dispatch(request, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/views/generic/base.py", line 98, in dispatch
    return handler(request, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/django/views/generic/edit.py", line 142, in post
    return self.form_valid(form)
           ^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/src/combo/combo/apps/assets/views.py", line 171, in form_valid
    ckeditor_upload_view.post(self.request)
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/ckeditor/views.py", line 50, in post
    backend.image_verify(upload)
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/ckeditor/image/pillow_backend.py", line 20, in image_verify
    Image.open(f).verify()
    ^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3288, in open
    im = _open_core(fp, filename, prefix, formats)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3275, in _open_core
    _decompression_bomb_check(im.size)
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3183, in _decompression_bomb_check
    raise DecompressionBombError(msg)
PIL.Image.DecompressionBombError: Image size (201640000 pixels) exceeds limit of 178956970 pixels, could be decompression bomb DOS attack.
```

Pour reproduire j'upload une ressources : gestion du portail/ressource/transférer (boutton en haut a droite) et j'envoie une image trop grande.

J'ai ajouté cette stack au ticket vu que ça me semble lié (les appels à get_thumbnail() de sorl se trouvent en partie dans le même fichier combo/combo/apps/assets/views.py)

#4 - 10 janvier 2024 18:10 - Yann Weber

Méthode pour reproduire :
- uploader une image trop grande dans les ressources (le logo par exemple)
- consulter l'url https://combo.dev.publik.love/assets/header:logo?height=100&width=100
- la requète va renvoyer une 302 menant vers une 404 (tentative de générer la miniature dans un cache ?)

Mais coté combo on récupère bien :

```
Getting thumbnail for file [assets/dec_bomb2_apvpdHJ.jpg] at [100x100]
Image size (201640000 pixels) exceeds limit of 178956970 pixels, could be decompression bomb DOS attack.
Traceback (most recent call last):
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/sorl/thumbnail/base.py", line 104, in get_thumbnail
    source_image = default.engine.get_image(source)
                   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/sorl/thumbnail/engines/pil_engine.py", line 84, in get_image
    return Image.open(buffer)
           ^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3288, in open
    im = _open_core(fp, filename, prefix, formats)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3275, in _open_core
    _decompression_bomb_check(im.size)
  File "/home/yann/envs/publik-env-py3/lib/python3.11/site-packages/PIL/Image.py", line 3183, in _decompression_bomb_check
    raise DecompressionBombError(msg)
PIL.Image.DecompressionBombError: Image size (201640000 pixels) exceeds limit of 178956970 pixels, could be decompression bomb DOS attack.
Remote file [assets/dec_bomb2_apvpdHJ.jpg] at [100x100] does not exist
```

#5 - 15 février 2024 11:46 - Yann Weber

- Lié à Development #86995: vérifier que les images transférées en ressources sont manipulables avec PIL ajouté

#6 - 19 février 2024 16:01 - Yann Weber

Après quelques recherches, c'est visiblement plus problématique que ça en à l'air : il n'est, à priori, pas possible de simplement attraper l'erreur : l'erreur est déjà attrapé par sorl (https://github.com/jazzband/sorl-thumbnail/blob/master/sorl/thumbnail/base.py#L104 la première ligne de la stacktrace est dans un try: catch Exception:)

On pourrait malgré tout utiliser la méthode exists() sur l'instance renvoyé par get_thumbnail et renvoyer une 5xx quand la miniature n'est pas générée (au lieu de la 302 menant à la 404). Mais je pense que ça continuera à remonter l'erreur dans sentry.

Est-ce que ça vaut le coup d'implémenter ça malgré la remontée de l'erreur dans sentry ? Et aussi malgré le fait qu'on ne peut pas vraiment afficher d'erreur à l'utilisateur (j'imagine que les miniatures sont chargées depuis des balises <img> et que si on renvoie du HTML il ne sera pas visible par l'utilisateur) ?

Sinon j'imagine que c'est le logger (appelé ici https://github.com/jazzband/sorl-thumbnail/blob/master/sorl/thumbnail/base.py#L106) qui provoque la remontée dans sentry ? Mais je ne crois pas qu'interdire à sorl de logger des erreurs soit une bonne idée.