
Mongrey

Release 0.4.4

2015-05-26 14:08:29

1	Resume	1
2	Features	3
3	Installation sample - Mongrey Serveur - Backend Sqlite	5
4	OS Compatibility - 64 bits (only)	7
5	Table of Contents	9
5.1	Téléchargements	9
5.2	Mongrey - Installation	10
5.3	Intégration	15
5.4	Mongrey Configuration	17
5.5	Sécurité	18
5.6	Migrations	18
5.7	Copies d'Ecran	19
5.8	Astuces	21
5.9	Codes et Messages d'erreurs	21
5.10	Les autres projets de Liste Grise	21
5.11	Références	25
5.12	Glossaire	25
6	Indices and tables	27

Resume

Version 0.4.4

Author

Last updated 2015-05-26 14:08:29

License BSD

Code <https://github.com/radical-software/mongrey>

Issues <https://github.com/radical-software/mongrey/issues>

Doc EN <http://mongrey.readthedocs.org/en/latest/>

Doc FR <http://mongrey.readthedocs.org/fr/latest/>

Features

- Greylist Server high performance
- Backends: MongoDB, PostgreSQL, MySQL, Sqlite
- No software dependencies (except Backend);
- **Configuration by Country, IP address, Network address, Email, Domain, Regex**
 - For every policy filter
 - For black and white lists
- **Optional filters:**
 - Relay deny control
 - Spoofing
 - Directory control DB, (SMTP, LDAP en cours..)
 - RBL
 - SPF
- WebUI (optional)
- REST API (in progress...)
- Cache with Memory or Redis

Installation sample - Mongrey Serveur - Backend Sqlite

```
$ curl -L http://download.mongrey.io/latest/mongrey-server-sqlite > /usr/local/bin/mongrey-server
$ chmod +x /usr/local/bin/mongrey-server
$ /usr/local/bin/mongrey-server --version
```

OS Compatibility - 64 bits (only)

- Ubuntu 14.04 (Trusty)
- Debian 8 (jessie)
- CentOS 7
- Fedora 20
- OpenSuse 13.1 (bottle)

Table of Contents

5.1 Téléchargements

5.1.1 Versions intégrales

Serveur [Mongrey Server All](#) (*Contient le serveur et les drivers pour tous les backends*)

Web [Mongrey Web All](#) (*Contient l'application Web et les drivers pour tous les backends*)

5.1.2 MongoDB

Requis :

- Linux Ubuntu 14.04+ ou CentOS 7 (64bits)
- MongoDB Server 2.6.5+

Serveur [Mongrey Server MongoDB](#)

Web UI [Mongrey Web MongoDB](#)

5.1.3 PostgreSQL

Requis :

- Linux Ubuntu 14.04+ ou CentOS 7 (64bits)
- PostgreSQL Server 9.1+

Serveur [Mongrey Server MongoDB](#)

Web UI [Mongrey Web MongoDB](#)

5.1.4 MySQL

Requis :

- Linux Ubuntu 14.04+ ou CentOS 7 (64bits)
- MySQL Server 5+

Serveur [Mongrey Server MySQL](#)

Web UI [Mongrey Web MySQL](#)

5.1.5 Sqlite

Serveur [Mongrey Server Sqlite](#)

Web UI [Mongrey Web Sqlite](#)

5.1.6 Migration

Outil de migration multi-backends

Migration [Mongrey Migration](#)

5.2 Mongrey - Installation

5.2.1 Introduction

Vous avez le choix entre plusieurs méthodes d'installation.

La plus simple étant la version *binnaire* adaptée au backend que vous utiliserez (MongoDB, PostgreSQL, ...).

Les binaires sont des versions compilés à l'aide de l'outil PyInstaller et n'ont aucunes dépendances externes.

Tous ces binaires ont été testés avec :

- Ubuntu 14.04 (Trusty) - 64 bits
- Debian 8 (jessie) - 64 bits
- CentOS 7 - 64 bits
- Fedora 20 - 64 bits
- OpenSuse 13.1 (bottle)
- Python 2.7.6

Le processus de génération des binaires est visible sur [Mongrey Build](#):

5.2.2 Exemples

Version en cours: 0.4.4

Remplacez `[RELEASE]` et `[BACKEND]` par la version et le backend souhaité.

- [http://download.mongrey.io/\[RELEASE\]/mongrey-server-\[BACKEND\]](http://download.mongrey.io/[RELEASE]/mongrey-server-[BACKEND])

Pour la version la plus récente avec un backend MongoDB, utilisez *latest*:

- <http://download.mongrey.io/latest/mongrey-server-mongo>

5.2.3 Mongrey Serveur

Sqlite (par défaut)

```
$ curl -L http://download.mongrey.io/latest/mongrey-server-sqlite > /usr/local/bin/mongrey-server
$ chmod +x /usr/local/bin/mongrey-server
$ /usr/local/bin/mongrey-server --version
```

Le plus simple pour la configuration initiale:

```
$ export MONGREY_SERVER_SETTINGS=/etc/mongrey/server.yml
$ export MONGREY_DB=sqlite:///var/lib/mongrey/mongrey.db

$ mkdir -vp /var/lib/mongrey

$ /usr/local/bin/mongrey-server config-install
```

Editez si nécessaire le fichier /etc/mongrey/server.yml

Affichez la configuration en cours:

```
$ /usr/local/bin/mongrey-server config
```

Configurez Postfix:

Note: Ajoutez warn_if_reject, devant le check_policy_service pour évaluez Mongrey sans risque.

Si Mongrey renvoi une action de rejet pour un mail, il sera transformé en simple warning dans les logs.

Le contenu et l'ordre des entrées dans smtpd_recipient_restrictions dépend de votre installation. L'exemple ci-après n'est qu'une des nombreuses possibilités de Postfix.

```
$ vi /etc/postfix/main.cf

smtpd_recipient_restrictions = reject_unauth_destination, warn_if_reject, check_policy_service inet:

$ postfix reload
```

Ou en mode commande:

```
$ postconf -e 'smtpd_recipient_restrictions = reject_unauth_destination, warn_if_reject, check_policy

$ postfix reload
```

Démarrez Mongrey:

```
$ /usr/local/bin/mongrey-server start
```

Pour un démarrage en mode background:

```
$ /usr/local/bin/mongrey-server --pid /var/run/mongrey-server.pid start &
```

Pour arrêter Mongrey:

```
$ kill -TERM `cat /var/run/mongrey-server.pid`
```

MongoDB

```
$ curl -L http://download.mongrey.io/latest/mongrey-server-mongo > /usr/local/bin/mongrey-server
$ chmod +x /usr/local/bin/mongrey-server
$ /usr/local/bin/mongrey-server --version
```

PostgreSQL

```
$ curl -L http://download.mongrey.io/latest/mongrey-server-postgresql > /usr/local/bin/mongrey-server
$ chmod +x /usr/local/bin/mongrey-server
$ /usr/local/bin/mongrey-server --version
```

MySQL

```
$ curl -L http://download.mongrey.io/latest/mongrey-server-mysql > /usr/local/bin/mongrey-server
$ chmod +x /usr/local/bin/mongrey-server
$ /usr/local/bin/mongrey-server --version
```

5.2.4 Docker

Docker - MongoDB

Installez la version binaire de mongrey pour *MongoDB*

```
$ docker pull mongo

# Lancement du server MongoDB
$ docker run --name mongodbl -d mongo mongod --smallfiles --directoryperdb

# Lancement de mongrey
$ docker run -it --rm --link mongodbl:mongodb \
  -e MONGREY_DB=mongodb://mongodb/mongrey \
  -e MONGREY_HOST=0.0.0.0 \
  -e MONGREY_PORT=9999 \
  -p 127.0.0.1:9999:9999 \
  -v /usr/local/bin/mongrey-server:/usr/local/bin/mongrey-server \
  ubuntu:14.04 /usr/local/bin/mongrey-server start
```

Docker - PostgreSQL

Installez la version binaire de mongrey pour *PostgreSQL*

```
$ docker pull postgres

# Lancement du server PostgreSQL
$ docker run --name pgsql1 -e POSTGRES_PASSWORD=secret -d postgres
```



```
# Création de la DB
$ docker exec -it pgsql1 sh -c 'exec psql -c "create database mongrey;" -U postgres'

# Lancement de mongrey
$ docker run -it --rm --link pgsql1:pgsql \
  -e MONGREY_DB=postgresql://postgres:secret@pgsql/mongrey \
  -e MONGREY_HOST=0.0.0.0 \
  -e MONGREY_PORT=9999 \
  -p 127.0.0.1:9999:9999 \
  -v /usr/local/bin/mongrey-server:/usr/local/bin/mongrey-server \
  ubuntu:14.04 /usr/local/bin/mongrey-server start
```

Docker - MySQL

Installez la version binaire de mongrey pour *MySQL*

```
$ docker pull mysql

# Lancement du server
$ docker run --name mysql1 -e MYSQL_ROOT_PASSWORD=secret -d mysql

# Création de la DB
$ docker exec -it mysql1 sh -c 'exec mysql -e "create database mongrey;" -uroot -p"secret"'

# Lancement de mongrey
$ docker run -it --rm --link mysql1:mysql \
  -e MONGREY_DB=mysql://root:secret@mysql/mongrey \
  -e MONGREY_HOST=0.0.0.0 \
  -e MONGREY_PORT=9999 \
  -p 127.0.0.1:9999:9999 \
  -v /usr/local/bin/mongrey-server:/usr/local/bin/mongrey-server \
  ubuntu:14.04 /usr/local/bin/mongrey-server start
```

Docker - Cache Redis

Pour utiliser un cache Redis dans mongrey, utiliser la variable d'environnement MONGREY_CACHE

```
$ docker pull redis

# Lancement du server
$ docker run -d --name redis1 -m 512m redis redis-server --appendonly yes
```

Exemple d'utilisation avec Mongrey dans Docker/Mongodb:

```
$ docker run -it --rm \
  --link mongodbl:mongodbl \
  --link redis1:redis \
  -e MONGREY_CACHE=redis://redis:6379/0 \
  -e MONGREY_DB=mongodb://mongodbl/mongrey \
  -e MONGREY_HOST=0.0.0.0 \
  -e MONGREY_PORT=9999 \
  -p 127.0.0.1:9999:9999 \
  -v /usr/local/bin/mongrey-server:/usr/local/bin/mongrey-server \
  ubuntu:14.04 /usr/local/bin/mongrey-server start
```

Docker - Build

Pour intégrer et personnaliser l'installation de Mongrey dans Docker, vous pouvez utiliser le modèle suivant.

Exemple avec la version binaire de Mongrey Server PostgreSQL:

```
# Dockerfile

FROM ubuntu:14.04

RUN apt-get update -y

RUN DEBIAN_FRONTEND=noninteractive \
    apt-get install -y --no-install-recommends \
    ca-certificates git curl language-pack-en

RUN curl -L http://download.mongrey.io/latest/mongrey-server-postgresql > /usr/local/bin/mongrey-server

RUN chmod +x /usr/local/bin/mongrey-server

CMD /usr/local/bin/mongrey-server start
```

```
# Build du contenair

$ docker build -t mongrey-server-postgresql .
```

5.2.5 Mongrey Web

Authentification

Le login et mot de passe de l'utilisateur par défaut sont: admin / mongrey

Vous pouvez les modifier par les variables d'environnement avant de lancer l'application:

```
MONGREY_WEB_USERNAME=monlogin

MONGREY_WEB_PASSWORD=monpassword
```

Pour créer le compte admin, utilisez la commande:

```
$ /usr/local/bin/mongrey-web default-user
```

MongoDB

```
$ curl -L http://download.mongrey.io/latest/mongrey-web-mongo > /usr/local/bin/mongrey-web-mongo

$ chmod +x /usr/local/bin/mongrey-web-mongo

$ /usr/local/bin/mongrey-web --help

$ /usr/local/bin/mongrey-web server -p 8081 -h 127.0.0.1

# Ouvrez le navigateur à l'adresse http://127.0.0.1:8081
```

PostgreSQL

```
$ curl -L http://download.mongrey.io/latest/mongrey-web-postgresql > /usr/local/bin/mongrey-web
$ chmod +x /usr/local/bin/mongrey-web
$ /usr/local/bin/mongrey-web --help
```

5.2.6 Installation par PIP

Requis :

- Linux récent (Ubuntu 14+, CentOS+)
- Librairie de compilation
- Python 2.7+ (python 3 non supporté pour l'instant)
- Python Setuptools/Pip
- Bibliothèques optionnelles selon le backend (mysql, postgresql, ...)

```
# Serveur et Web - Tous les backend
$ pip install mongrey[full]

# Serveur - Backend MongoDB
$ pip install mongrey[server_mongodb]

# Serveur - Tous les backend
$ pip install mongrey[server]

$ mongrey-server --help
$ mongrey-web --help
```

5.2.7 APT

Todo

5.2.8 RPM

Todo

5.3 Intégration

Note: Mongrey WEB est livré avec un serveur WSGI [Gevent](#) intégré très performant mais si vous avez besoin d'intégrer l'application dans Nginx ou Apache, vous pouvez suivre les procédures suivantes.

5.3.1 Mongrey Serveur

5.3.2 Mongrey WEB - WSGI

Todo

A tester

Warning: Le téléchargement de Mongrey Web au format binaire n'est pas compatible avec cette installation. Il faut installer Mongrey Web normalement en utilisant la procédure par *Installation par PIP*

Gunicorn

```
$ pip install gunicorn  
$ gunicorn -k gevent_wsgi --workers 1 'mongrey.web.wsgi:create_app()'
```

Gunicorn - Supervisord

```
[program:mongrey-web]  
command=gunicorn -k gevent_wsgi --workers 1 'mongrey.web.wsgi:create_app()'  
autostart=true  
autorestart=true  
redirect_stderr=True  
stdout_logfile=/var/log/supervisor/%(program_name)s.log  
stderr_logfile=/var/log/supervisor/%(program_name)s.log
```

Chaussette

Todo

5.3.3 Upstart

<http://upstart.ubuntu.com/>

Todo

5.3.4 Systemd

<http://freedesktop.org/wiki/Software/systemd/>

Todo

5.4 Mongrey Configuration

5.4.1 Mongrey Serveur

Configuration par défaut

Par défaut, Mongrey Serveur utilise les variables d'environnements mais vous pouvez aussi charger la configuration à partir d'un fichier YAML.

La commande **mongrey-server install** créer un fichier YAML avec la configuration par défaut.

A chaque chargement, Mongrey Serveur, cherche un fichier de configuration facultatif:

Note:

- **Ordre de recherche:**
 - Valeur de l'option `mongrey-server --settings myfile.yml`
 - Valeur de la variable d'environnement `MONGREY_SERVER_SETTINGS`
 - `/etc/mongrey/server.yml`
 - `~/mongrey/server.yml`
-

Note: Le fichier YAML n'a pas besoin d'être rempli complètement

Après son chargement, les valeurs sont concaténés avec la configuration par défaut (variables d'environnements)

5.4.2 Postfix intégration

- Voir [Postfix_Policy](#) pour plus de détail

```
#/etc/postfix/main.cf:

smtpd_recipient_restrictions =
    ...
    reject_unauth_destination
    check_policy_service inet:127.0.0.1:9999
```

5.4.3 Politique de liste grise

Todo

5.4.4 Liste Blanches

Todo

Liste Blanches

5.4.5 Géo-localisation

Todo

5.4.6 Greylisting

Todo

5.5 Sécurité

5.5.1 Mongrey Serveur - IP Autorisés

Défaut: 127.0.0.1, ::1

Par défaut, Mongrey Serveur n'accepte des connections qu'en provenance de l'ip loopback (127.0.0.1, ::1)

Toutes autres connection est rejetés et une entrée de log permet à un outil comme [Fail2ban](#) d'agir dynamiquement sur le firewall.

Si [Fail2ban](#) est activé et correctement paramètré, les prochaines tentatives seront arrêtés par le firewall.

5.5.2 Mongrey Web - IP Autorisés

Défaut: Toutes

Si vous utilisez le serveur WSGI intégré à Mongrey Web, vous pouvez activer la même protection par IP que Mongrey Serveur.

5.5.3 Fail2ban et iptables

[Fail2ban](#) est un outil Python qui à partir de règles et des logs, permet de créer dynamiquement des règles de blocage dans le firewall.

C'est l'outil idéal pour combattre les attaques par force brute.

5.6 Migrations

5.6.1 Radical-Spam

Le but de cette migration n'est pas de remplacer RadicalSpam mais seulement la partie Greylisting ainsi que certaines fonctionnalités assurés actuellement par Postfix à travers des fichiers plats.

Données gérés par l'outil de migration:

- Domaines internet (local-relays)
- Autorisations Mynetwork (local-mynetwork-*)
- Listes Noires (local-blacklist-*)

- Mailbox (local-directory)

5.6.2 Exemple de Migration Radical-Spam avec Mongrey Serveur - Sqlite

New in version 0.4.2.

Cette opération va récupérer les Domaines, Mynetwork, l'annuaire et toutes les liste noires de Radical-Spam et les enregistrer dans Mongrey.

Installation

```
$ curl -L http://download.mongrey.io/latest/mongrey-migration > /usr/local/bin/mongrey-migration
$ chmod +x /usr/local/bin/mongrey-migration
$ /usr/local/bin/mongrey-migration --version
```

Migration

```
# simulation (-n)
$ /usr/local/bin/mongrey-migration -P /var/rs/addons/postfix/etc -n radicalspam

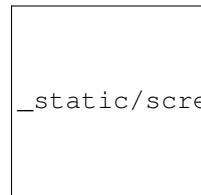
# Migration:
$ /usr/local/bin/mongrey-migration -P /var/rs/addons/postfix/etc radicalspam
```

5.6.3 Postgrey

Todo

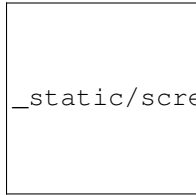
5.7 Copies d'Ecran

5.7.1 Authentification par formulaire



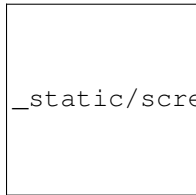
_static/screenshots/login.png

5.7.2 Liste grise



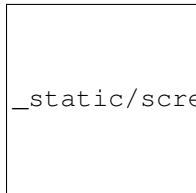
`_static/screenshots/greylists-list.png`

5.7.3 Détail d'une entrée de liste grise



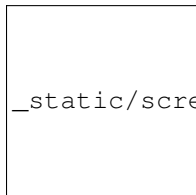
`_static/screenshots/greylists-show-light.png`

5.7.4 Liste blanche



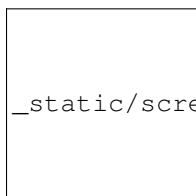
`_static/screenshots/whitelist-list.png`

5.7.5 Liste blanche - Domaine destinataire



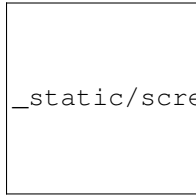
`_static/screenshots/whitelist-form-domain.png`

5.7.6 Liste blanche - Hostname par Expression régulière



`_static/screenshots/whitelist-form-regex.png`

5.7.7 Liste blanche - Adresse IP Réseau (cidr)



_static/screenshots/whitelist-form-ip-cidr.png

5.8 Astuces

5.8.1 Adresses IP variables

5.9 Codes et Messages d'erreurs

5.9.1 554

554 5.7.1 blacklisted

554 5.7.1 relay denied

554 5.7.1 spoofing

5.9.2 greylisted

Exemple: 450 4.2.0 Greylisted for 30 seconds

5.10 Les autres projets de Liste Grise

5.10.1 Postgrey

Url Postgrey

Mise à jour Juin 2014

Techno Perl

DB Supportées ?

Web UI ?

Licence ?

5.10.2 Bley

Url Bley

Mise à jour Mai 2015

Techno Python (Twisted Framework)

DB Supportées ?

Web UI ?

Licence ?

5.10.3 Valvula

Url <http://www.aspl.es/valvula/>

Mise à jour ?

Techno C

DB Supportées ?

Web UI <http://www.core-admin.com/portal/> (payante)

Licence GPL 2.0

5.10.4 policyd-weight

Url <http://www.policyd-weight.org/>

Mise à jour Septembre 2011 (maintenance mode)

Techno Perl

DB Supportées ?

Web UI Non

Licence ?

5.10.5 Gross

Url <https://code.google.com/p/gross/>

Mise à jour Juin 2011

Techno C

DB Supportées ?

Web UI ?

Licence ?

5.10.6 pfxpold

Url <https://github.com/mschmitt/pfxpold>

Mise à jour Juin 2010

Techno Perl

DB Supportées ?

Web UI ?

Licence ?

5.10.7 policyd

Url <http://wiki.policyd.org/download>

Mise à jour Octobre 2013

Techno Perl

DB Supportées ?

Web UI ?

Licence ?

5.10.8 ACL Policy

Url <http://www.apolicy.org/>

Mise à jour Septembre 2008

Techno Python (Twisted Framework)

DB Supportées ?

Web UI ?

Licence ?

5.10.9 Ppolicy

Url <http://kmlinux.fjfi.cvut.cz/~vokacpet/activities/ppolicy/>

Mise à jour Décembre 2008

Techno Python (Twisted Framework)

DB Supportées ?

Web UI ?

Licence ?

5.10.10 Tumgreyspf

Url <https://github.com/linsomniac/tumgreyspf>

Mise à jour Avril 2015

Techno Python

DB Supportées ?

Web UI ?

Licence ?

5.10.11 Greyfix

Url <http://www.kim-minh.com/pub/greyfix/>

Mise à jour Mars 2013

Techno C

DB Supportées ?

Web UI ?

Licence ?

5.10.12 Sqlgrey (fork de Postgrey)

Url <http://sqlgrey.sourceforge.net/>

Mise à jour Février 2012

Techno Perl

DB Supportées ?

Web UI ?

Licence ?

5.10.13 Greylist Daemon

Url <http://mimo.gn.apc.org/greylist>

Mise à jour Octobre 2009

Techno C++

DB Supportées ?

Web UI ?

Licence ?

5.10.14 Gld

Url <https://www.gasmi.net/progs.php>

Mise à jour Mai 2006

Techno C

DB Supportées ?

Web UI ?

Licence ?

5.11 Références

5.12 Glossaire

Liste Grise

- Définition par Wikipédia [Greylisting](#)

binaire, Version binaire Pour faciliter l'utilisation de Mongrey, des binaires autonomes sont livrés pour chaque déclinaisons de Mongrey

Chaque binaire est préparé à l'aide de [PyInstaller](#) et n'a besoin d'aucune installation supplémentaire, en dehors du backend (MongoDB, PostgreSQL ou MySQL).

Indices and tables

- `genindex`
- `search`

B

binaire, [25](#)

L

Liste Grise, [25](#)

V

Version binaire, [25](#)