
s3recon Documentation

Release 1.4.0

Travis Clarke

Mar 22, 2020

Contents

1	Installation	3
2	Usage	5
3	Example 1: Output to a json file	7
4	Example 2: Output to a MongoDB database	9
5	FAQ	11
6	Going Forward	13
7	Disclaimer	15
8	License	17

Version 1.4.0

Amazon S3 bucket finder and crawler.

CHAPTER 1

Installation

NOTE: s3recon requires python version **>=3.6**

```
$ pip install s3recon
```


CHAPTER 2

Usage

```
usage: s3recon [-h] [-o file] [-d] [-p] [-t seconds] [-v] [-c num] word_list [word_
↳list ...]

positional arguments:
  word_list              read words from one or more <word-list> files

optional arguments:
  -h, --help            show this help message and exit
  -o file, --output file write output to <file>
  -d, --db              write output to database
  -p, --public          only include 'public' buckets in the output
  -t seconds, --timeout seconds http request timeout in <seconds> (default: 30)
  -v, --version         show program's version number and exit
  -c num, --concurrency num maximum <num> of concurrent requests (default: # of_
↳lcpus)
```

Example 1: Output to a json file

3.1 1. Download a word-list.

The [SecLists](#) repository has a multitude of word-lists to choose from. For this example, let's download the sample word-list included in this repository.

```
$ curl -sSfL -o "word-list.txt" "https://raw.githubusercontent.com/clarketm/s3recon/  
↪master/data/words.txt"
```

3.2 2. Run s3recon.

Execute `s3recon` using the `word-list.txt` file and output the public S3 buckets to a json file named `results.json`.

```
$ s3recon "word-list.txt" -o "results.json" --public  
  
- PRIVATE https://s3.sa-east-1.amazonaws.com/test-lyft  
- PRIVATE https://s3.ap-south-1.amazonaws.com/test.amazon  
+ PUBLIC https://walmart-dev.s3.us-east-1.amazonaws.com  
- PRIVATE https://s3.ap-southeast-1.amazonaws.com/apple-prod  
- PRIVATE https://walmart.s3.ap-southeast-1.amazonaws.com  
...
```

3.3 3. Inspect the results.

Check the `results.json` output file to view the S3 buckets you have discovered!

```
$ cat "results.json"
```

```
{
  "public": {
    "total": 12,
    "hits": [
      "https://walmart-dev.s3.us-east-1.amazonaws.com",
      "https://apple-production.s3.ap-southeast-1.amazonaws.com",
      ...
    ]
  }
}
```

Note: to include `private` buckets in the results omit the `-p`, `--public` flag from the command.

3.4 4. Crawl the results.

Enumerate the static files located in each bucket and record the findings.

Coming soon!

Example 2: Output to a MongoDB database

4.1 1. Download a word-list.

The [SecLists](#) repository has a multitude of word-lists to choose from. For this example, let's download the sample word-list included in this repository.

```
$ curl -sSfL -o "word-list.txt" "https://raw.githubusercontent.com/clarketm/s3recon/  
↪master/data/words.txt"
```

4.2 2. Start an instance of MongoDB

```
$ docker run --name "mongo" -p 27017:27017 -v "mongodb_data:/data/db" -v "mongodb_  
↪config:/data/configdb" -d mongo
```

4.3 3. Run s3recon.

Execute `s3recon` using the `word-list.txt` file and output to MongoDB instance.

```
$ s3recon "word-list.txt" --db  
  
- PRIVATE https://s3.sa-east-1.amazonaws.com/test-lyft  
- PRIVATE https://s3.ap-south-1.amazonaws.com/test.amazon  
+ PUBLIC https://walmart-dev.s3.us-east-1.amazonaws.com  
- PRIVATE https://s3.ap-southeast-1.amazonaws.com/apple-prod  
- PRIVATE https://walmart.s3.ap-southeast-1.amazonaws.com  
...
```

4.4 3. Inspect the results.

Check the MongoDB database: s3recon collection: hits to view the S3 buckets you have discovered!

```
$ mongo "s3recon" --quiet --eval 'db.hits.find({}, {"url": 1, "access": 1, "_id": 0}).
↪ limit(5) '
```

```
{ "url" : "https://s3.us-east-2.amazonaws.com/apple", "access" : "private" }
{ "url" : "https://s3.us-west-1.amazonaws.com/microsoft-dev", "access" : "private" }
{ "url" : "https://s3.us-west-1.amazonaws.com/dev-microsoft", "access" : "private" }
{ "url" : "https://s3.us-east-2.amazonaws.com/amazon", "access" : "private" }
{ "url" : "https://s3.us-east-1.amazonaws.com/dev-amazon", "access" : "private" }
```

4.5 4. Crawl the results.

Enumerate the static files located in each bucket and record the findings.

Coming soon!

5.1 Q: How do I configure this utility?

5.2 A:

`s3recon` can be configured using a `yaml` configuration file located in either the current working directory (e.g. `./s3recon.yml`) or your home directory (e.g. `~/s3recon.yml`).

The following is the list of configurable values:

```
# s3recon.yml

database: { host: "0.0.0.0", ... }

separators: ["-", "_", "."]

environments: ["", "backup", "backups", ...]

regions: ["ap-northeast-1", "ap-northeast-2", ...]
```

To see the full list of configurable values (and their **defaults**) please refer to the `s3recon.yml` file in this repository.

5.3 Q: How do I customize the AWS regions used in the recon?

5.4 A:

The AWS *regions* can be altered by setting the `regions` array in your `s3recon.yml` configuration file.

```
# s3recon.yml
regions: [ "us-west-2", ...]
```

5.5 Q: How do I customize the environment values used in the recon?

5.6 A:

The *environments* are modifiers permuted with each item of the *word-list* (and the *separator*) to construct the bucket value in request. The value can be altered by setting the `environments` array in your `s3recon.yml` configuration file.

For example, to only search lines from the word-list *verbatim* (i.e. without modification) you can set this value to an empty array.

```
# s3recon.yml
environments: []
```

5.7 Q: How do I customize the MongoDB host and port?

5.8 A:

The database *host* and *port* can be configured by altering the database map in your `s3recon.yml` configuration file.

For example, `host` and `port` can be set directly inside the database map

```
# s3recon.yml
database: {
  host: "0.0.0.0",
  port: 27017
}
```

5.9 Q: How do I use a database other than MongoDB?

5.10 A:

Sorry, at the moment only MongoDB is supported.

CHAPTER 6

Going Forward

- ☐ Create `crawl` command to crawl public/private buckets found in `find` stage.
- ☐ Separate out `find` and `crawl` as subcommands.
- ☒ Store discovered buckets in a NoSQL database.

CHAPTER 7

Disclaimer

This tools is distributed for educational and security purposes. I take no responsibility and assume no liability for the manner in which this tool is used.

CHAPTER 8

License

MIT © Travis Clarke