**GitHub dorks**

This is the oldest and most traditional way to access sensitive data from public repositories, and because it’s a part of public sources, it can be included in any OSINT research.

What’s needed? Only GitHub account and some basic knowledge about programming variables; in other words, the things you’ll be looking for, such as database user and password, secret access keys, tokens, etc.

Here at SecurityTrails, we wanted to see how much time an average user takes to locate sensitive data from public repositories, so we tested a few random sensitive keywords such as:

* password
* dbpassword
* dbuser
* access\_key
* secret\_access\_key
* bucket\_password
* redis\_password
* root\_password

|  |  |
| --- | --- |
| **Search Term** | **Result Expected** |
| “Company” security\_credentials | LDAP (active directory) |
| “Company” connectionstring | Database Credentials |
| “Company” JDBC | Database Credentials |
| “Company” ssh2\_auth\_password | Unauthorized access to servers |
| “Company”send\_keys or send,keys | If other keywords related to password failed |
| “Company” Language:Python password | Password if language used Python in Org. |

It took less than 60 seconds to find the first results of GitHub repositories leaking data about database connections — including usernames and passwords — as you can see below:



There’s an even more effective approach to take, by searching for specific credentials strings in configuration files, such as this example: filename:sftp-config.json password.

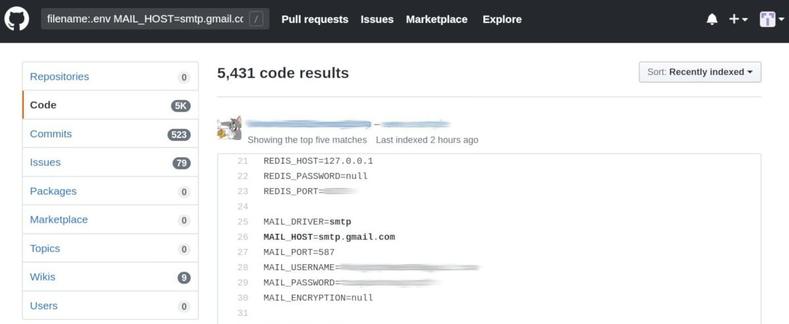


As shown, we were able to obtain the IP address, root password and SSH port credentials.

Another tip that may increase your success in finding leaked data is ordering the results, by setting the GitHub filter to ‘Recently indexed’, as you see below:

Recently indexed

What about smtp login credentials? It’s easy by searching for strings like: filename:.env MAIL\_HOST=smtp.gmail.com. Output example:



How about SQL dumps? Simply use extension:sql mysql dump, and the results will appear before your very eyes:



There are numerous GitHub dorks that can be used to scan GitHub repositories, including:

* filename:.npmrc \_auth
* filename:.dockercfg auth
* extension:pem private
* extension:ppk private
* filename:id\_rsa or filename:id\_dsa
* extension:sql mysql dump
* extension:sql mysql dump password
* filename:credentials aws\_access\_key\_id
* filename:.s3cfg
* filename:wp-config.php
* filename:.htpasswd
* filename:.env DB\_USERNAME NOT homestead
* filename:.env MAIL\_HOST=smtp.gmail.com
* filename:.git-credentials

Link: <https://github.com/random-robbie/keywords/blob/master/keywords.txt>