Jerry Shah (Jerry)

1.3K Followers

GitHub Recon - It's Really Deep

Jerry Shah (Jerry) Jul 9, 2020 · 6 min read

Hello everyone, I know that my speed of writing blogs has been decreased it's because I'm busy with some other stuff. Doesn't matter I have came up with this great blog as a part of **recon** because everywhere recon is important and I hope you guys will like it.

Summary :

Everyone knows what a github is but let me give you a brief about it.

What is GitHub ?

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, **GitHub** provides a Web-based graphical interface.

Apart from this it also contains API keys, passwords, customer data etc. Basically it contains a lot of sensitive information which can be useful for an attacker. This sensitive information leaks can cost a company thousand dollars of damage. Let's see the basic concept first of github recon.

We will be covering two ways of github recon :

1. Manual (Code Search OR GitHub Dorking)

2. Automated (Using Tools)

1. **Manual - Code Search || GitHub Dorking**

Code search is nothing but the use of some keywords that helps you to find sensitive things like passwords, API keys, database files etc.

You can search for code globally across all of GitHub. You can also search for code within a particular repository or organization. To search for code across all public repositories, you must be signed in to a GitHub account. GitHub provides "rich code searching" that scans public github repositories.

GitHub Search

GitHub Search

How to do a recon on GitHub ?

1. You can use simple queries like facebook.com or google.com etc. to search for a particular company.

Facebook Search

2. You can also use multi-word strings like **“Authorization: Bearer”**

Multi-word String

Now you need to open a repository and have to search for the authorization token or password or any other sensitive information.

Password (Secret)

3. You can search for specific filenames like **“filename:vim_settings.xml”**

4. You can search for specific languages like “language:PHP”

Specific Language

This was the basics of the github dorking but you can also combine your queries like **“facebook.com filename:vim_settings.xml”** which will give you all the vim_settings.xml file of a particular company facebook. Same way you can also perform different query searches.

It’s not easy to find sensitive information on github you need to spend a lot of time and need to check for each repository of a particular company, so their is this concept of **GitHub Dorking** which reduces your effort of searching sensitive information manually.

Apart from repositories you can also check for **code**, commits, issues, discussions, packages, marketplace, **topics**, **wikis** and **users.**

Other searches

Apart from using GitHub Dorks, you can directly search for the source. For doing that you need to find the your target company’s github page and from there you can find all their developers and monitor their accounts.

Company's Github Page

Once you find your target company's github page you just need to check the list of people that are associated with your target company. This can be done by clicking on the "people" tab.

Target Company's People

Now you will need to manually go through each one and look for exposures and this will take long time. You should be looking for urls, api keys, usernames, passwords etc. It might be possible that someone has uploaded something sensitive here.

GitHub Dork List :

GitHub Dorks for Finding Files

filename:manifest.xml
filename:travis.yml
filename:vim_settings.xml
filename:database
filename:prod.exs NOT prod.secret.exs
filename:prod.secret.exs
filename:.npmrc _auth
filename:.dockercfg auth
filename:WebServers.xml
filename:.bash_history <Domain name>
filename:sftp-config.json
filename:sftp.json path:.vscode
filename:secrets.yml password
filename:.esmtprc password
filename:passwd path:etc
filename:dbeaver-data-sources.xml
path:sites databases password
filename:config.php dbpasswd
filename:prod.secret.exs
filename:configuration.php JConfig password

filename:.sh_history
shodan_api_key language:python
filename:shadow path:etc
JEKYLL_GITHUB_TOKEN
filename:proftpdpasswd
filename:.pgpass
filename:idea14.key
filename:hub oauth_token
HEROKU_API_KEY language:json
HEROKU_API_KEY language:shell
SF_USERNAME salesforce
filename:.bash_profile aws
extension:json api.forecast.io
filename:.env MAIL_HOST=smtp.gmail.com
filename:wp-config.php
extension:sql mysql dump
filename:credentials aws_access_key_id
filename:id_rsa or filename:id_dsa

GitHub Dorks for Finding Languages

language:python username
language:php username
language:sql username
language:html password
language:perl password
language:shell username
language:java api
HOMEBREW_GITHUB_API_TOKEN language:shell

GiHub Dorks for Finding API Keys, Tokens and Passwords

api_key
“api keys”
authorization_bearer:
oauth
auth
authentication
client_secret
api_token:
“api token”
client_id
password
user_password
user_pass
passcode
client_secret
secret
password hash
OTP
user auth

GitHub Dorks for Finding Usernames

user:name (user:admin)
org:name (org:google type:users)
in:login (<username> in:login)
in:name (<username> in:name)
fullname:firstname lastname (fullname:<name> <surname>)
in:email (data in:email)

GitHub Dorks for Finding Information using Dates

created:<2012–04–05
created:>=2011–06–12
created:2016–02–07 location:iceland
created:2011–04–06..2013–01–14 <user> in:username

GitHub Dorks for Finding Information using Extension

extension:pem private
extension:ppk private
extension:sql mysql dump
extension:sql mysql dump password
extension:json api.forecast.io
extension:json mongolab.com
extension:yaml mongolab.com
[WFClient] Password= extension:ica
extension:avastlic “support.avast.com”
extension:json googleusercontent client_secret

So this was all about manual technique to find sensitive information on github, lets move to some automated technique.

2. **Automated Technique - Using Tools**

However automation makes the process easy and fast but it also has it’s own drawback of false-positive results. Not every time the result is false-positive but sometimes it may happen. I have some automated tools that will help you to find sensitive things on github.

1. **TruggleHog :**

It is easy to use. It searches through git repositories for secrets, digging deep into commit history and branches. This is effective at finding secrets accidentally committed.

How to use it ?

1. Go to https://github.com/dxa4481/truffleHog and clone it (download it)
2. Use to below given command to find for sensitive information

Command : python3 trufflehog.py --regex --entropy=False https://github.com/<yourTargetRepo>

TruffleHog

2. **Github-Dorks :**

It is a simple python tool that can search through your repository or your organization/user repositories.

How to use it ?

1. Go to https://github.com/techgaun/github-dorks and clone it (download it)
2. Install all the given requirements
3. Use the below given command to search for all the repositories of a single user

Command : python github-dork.py -u <username>

Github-Dork.py

User Information Leak

3. **Watchtower :**

AI-powered scanner to detect API keys, secrets, sensitive information. Watchtower Radar API lets you integrate with GitHub public or private repository, AWS, GitLab, Twilio, etc. The scan results are available on a web interface or CLI output. You can read more about

it here : https://radar.nightfall.ai/docs#get-results. Basically it is a web application that helps you to scan github repositories.

How to use it ?

1. Go to https://radar.nightfall.ai/ and login with your github account.
2. Simply add your github's target URL on the left top section for scanning

Scanning Target

3. After the scan is completed click on results to view the information and you'll be redirected to another page like below one

Scan Completed

4. Now click on GitHub to see the leaked information on github

Result

Some other automated tools for scanning GitHub Repositories :

https://github.com/BishopFox/GitGot
https://github.com/Talkaboutcybersecurity/GitMonitor
https://github.com/michenriksen/gitrob
https://github.com/tillson/git-hound
https://github.com/kootenpv/gittyleaks
https://github.com/awslabs/git-secrets https://git-secret.io/

NOTE : If you find any API key or credentials or any other sensitive information under test directory then do not report it because that is an intended behaviour.

Bug Bounty | Hall of Fame | Vulnerability | Hackerone | Bugcrowd