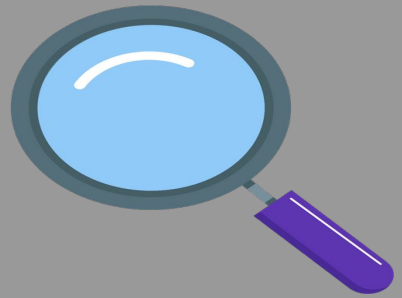
# GitHub

## Sensitive Information



**Mahmoud M. Awali**

@0xAwali

# Employees of The Company

- **Google**
  Search Engine

Google

site:github.com inurl:"org=company"

- **GitHub**

org:company

github.com

Search Github

# .docker/.dockercfg OR .docker/config.json
# Docker Registry Authentication Data

- File

org:company filename:.dockercfg

org:company docker AND auth AND email

user:name filename:.dockercfg

user:name docker AND auth AND email

"company.com" docker AND auth AND email

github.com

Search Github

# .idea/webservers.xml
# Contains Web Server Credentials

- File

org:company filename:webservers.xml

org:company fileTransfer AND pass

user:name filename:webservers.xml

user:name fileTransfer AND pass

"company.com" fileTransfer AND pass

github.com

Search Github

# .mozilla/firefox/logins.json
# Firefox Saved Password Collection

- File

org:company filename:firefox/logins.json

org:company encryptedUsername encryptedPassword

user:name filename:firefox/logins.json

user:name encryptedUsername encryptedPassword

"company.com" encryptedUsername encryptedPassword

github.com

Search Github

# .ssh/id_rsa OR .ssh/id_rsa.pub
# Private SSH Key

- File

org:company filename:.ssh/id_rsa

org:company "BEGIN RSA PRIVATE KEY" OR ssh-rsa

user:name filename:.ssh/id_rsa

user:name "BEGIN RSA PRIVATE KEY" OR ssh-rsa

"company.com" "BEGIN RSA PRIVATE KEY" OR ssh-rsa

github.com

Search Github

# .vscode/sftp.json

# Contains SFTP OR SSH Server Credentials

- File

org:company filename:sftp.json

org:company host AND pass

user:name filename:sftp.json

user:name host AND pass

"company.com" host AND pass

github.com

Search Github

# cloud/.credentials
# Contains S3 Credentials

- File

org:company filename:.credentials

org:company aws_secret_access_key OR aws_secret_key

user:name filename:.credentials

user:name aws_secret_access_key OR aws_secret_key

"company.com" aws_secret_access_key aws_secret_key

github.com

Search Github

# cloud/.s3cfg
# Contains S3 Credentials

- File

org:company filename:.s3cfg

org:company secret_key

user:name filename:.s3cfg

user:name secret_key

"company.com" secret_key

github.com

Search Github

# cloud/.tugboatls

# Digital Ocean Tugboat Config

- File

org:company filename:.tugboat

org:company authentication AND api_key

user:name filename:.tugboat

user:name authentication AND api_key

"company.com" authentication AND api_key

Search Github

# cloud/heroku.json
# Contains Heroku Config

- File

org:company filename:heroku.json

org:company HEROKU_API_KEY OR HEROKU_KEY

user:name filename:heroku.json

user:name HEROKU_API_KEY OR HEROKU_KEY

"company.com" HEROKU_API_KEY OR HEROKU_KEY

github.com

Search Github

# db/.pgpass
# PostgreSQL File Contains Passwords

- File

org:company filename:.pgpass

org:company :database:

user:name filename:.pgpass

user:name :database:

"company.com" :database:

github.com

Search Github

# db/dbeaver-data-sources.xml
# DBeaver Config Containing Credentials

- File

org:company filename:dbeaver-data-sources.xml

org:company connection AND jdbc

user:name filename:dbeaver-data-sources.xml

user:name connection AND jdbc

"company.com" connection AND jdbc

github.com

Search Github

# db/dump.sql
# Contains MYSQL Hashes Dump

- File

org:company filename:dump.sql

org:company "MySQL dump" AND "INSERT INTO"

user:name filename:dump.sql

user:name "MySQL dump" AND "INSERT INTO"

"company.com" "MySQL dump" AND "INSERT INTO"

Search Github

# db/mongoid.yml
# Mongoid Config File

- File

org:company filename:mongoid.yml

org:company production AND mongodb

user:name filename:mongoid.yml

user:name production AND mongodb

"company.com" production AND mongodb

github.com

Search Github

# db/robomongo.json

# Mongolab Credentials For Robomongo

- File

org:company filename:robomongo.json

org:company userPassword AND serverHost

user:name filename:robomongo.json

user:name userPassword AND serverHost

"company.com" userPassword AND serverHost

github.com

Search Github

# filezilla/filezilla.xml OR filezilla/recentservers.xml
# Filezilla config file

- File

org:company filename:filezilla.xml

org:company FileZilla3 AND "Pass encoding"

user:name filename:filezilla.xml

user:name FileZilla3 AND "Pass encoding"

"company.com" FileZilla3 AND "Pass encoding"

github.com

Search Github

# cert-key.pem
# PEM Private key

- File

org:company filename:cert-key.pem

org:company "BEGIN PRIVATE KEY"

user:name filename:cert-key.pem

user:name "BEGIN PRIVATE KEY"

"company.com" "BEGIN PRIVATE KEY"

github.com

Search Github

# Putty-example.ppk
# PuTTYgen Private Key

- File

org:company filename:putty extension:ppk

org:company PuTTY-User-Key-File

user:name filename:putty extension:ppk

user:name PuTTY-User-Key-File

"company.com" PuTTY-User-Key-File

github.com

Search Github

# django/settings.py

# Contains Valid Secret Key For Django Setup

- File

org:company filename:settings.py

org:company SECRET_KEY

user:name filename:settings.py

user:name SECRET_KEY

"company.com" SECRET_KEY

github.com

Search Github

# salesforce.js

# Salesforce Credentials In A NodeJS Project

- File

org:company filename:salesforce.js

org:company "conn.login(" OR "require('jsforce')"

user:name filename:salesforce.js

user:name "conn.login(" OR " require('jsforce')"

"company.com" "conn.login(" OR " require('jsforce')"

github.com

Search Github

# secrets.yml
# Contains Credentials For Ruby on Rails

- File

org:company filename:secrets.yml

org:company secret_key_base

user:name filename:secrets.yml

user:name secret_key_base

"company.com" secret_key_base

github.com

Search Github

# credentials.yml

# Credentials For Voice AND Chat Platforms

- **Mine**

`org:company filename:credentials.yml`

`org:company slack_token OR access-token OR _TOKEN`

`user:name filename:credentials.yml`

`user:name slack_token OR  access-token OR _TOKEN`

`"company.com" slack_token OR  access-token OR _TOKEN`

Search Github

# .travis.yml
# Contains The Slack Token For RocketChat

- Writeup

org:company filename:.travis.yml

org:company notification AND slack OR secure

user:name filename:.travis.yml

user:name notification AND slack OR secure

"company.com" notification AND slack OR secure

github.com

Search Github

# .env
# Contains Credentials For Laravel

- File

org:company filename:.env

org:company APP_KEY OR DB_PASS

user:name filename:.env

user:name APP_KEY OR DB_PASS

"company.com" APP_KEY OR DB_PASS

github.com

Search Github

# config.php
# PHP Application Configuration File

- File

org:company filename:config.php

org:company mysql_connect OR dbpass

user:name filename:config.php

user:name mysql_connect OR dbpass

"company.com" mysql_connect OR dbpass

github.com

Search Github

# wp-config.php
# WordPress Configuration File

- File

org:company filename:wp-config.php

org:company DB_PASSWORD OR AUTH_KEY

user:name filename:wp-config.php

user:name DB_PASSWORD OR AUTH_KEY

"company.com" DB_PASSWORDt OR AUTH_KEY

github.com

Search Github

# .esmtprc
# ESMTP Configuration File

- File

org:company filename:.esmtprc

org:company "hostname smtp"

user:name filename:.esmtprc

user:name "hostname smtp"

"company.com" "hostname smtp"

github.com

Search Github

# .ftpconfig

# Contains SFTP OR SSH Server Credentials

- File

org:company filename:.ftpconfig

org:company protocol AND ftp AND pass

user:name filename:.ftpconfig

user:name protocol AND ftp AND pass

"company.com" protocol AND ftp AND pass

github.com

Search Github

# .netrc
# Contains SMTP Credentials

- File

org:company filename:.netrc

org:company machine AND login AND password

user:name filename:.netrc

user:name machine AND login AND password

"company.com" machine AND login AND password

github.com

Search Github

# .npmrc
# NPM Registry Authentication Data

- File

```
org:company filename:.npmrc
org:company registry AND _auth OR _authToken
user:name filename:.npmrc
user:name registry AND _auth OR _authToken
"company.com" registry AND _auth OR _authToken
```

github.com

Search Github

# .remote-sync.json

# Contains FTP , SFTP OR SSH Credentials

- File

org:company filename:.remote-sync.json

org:company "remote sync" AND pass

user:name filename:.remote-sync.json

user:name "remote sync" AND pass

"company.com" "remote sync" AND pass

github.com

Search Github

# .config
# IRC Configuration

- File

org:company filename:config

org:company IRC_HOST AND IRC_PASS

user:name filename:config

user:name IRC_HOST AND IRC_PASS

"company.com" IRC_HOST AND IRC_PASS

github.com

Search Github

# deployment-config.json OR sftp-config.json
# Contains Server Details AND Credentials

- File

org:company filename:deployment-config.json

org:company type AND sftp AND pass

user:name filename:deployment-config.json

user:name type AND sftp AND pass

"company.com" type AND sftp AND pass

Search Github

# hub

# Hub Configuration That Stores Github Tokens

- File

org:company filename:hub oauth_token

org:company github.com AND user AND oauth_token

user:name filename:hub oauth_token

user:name github.com AND user AND oauth_token

"company.com" github.com AND user AND oauth_token

github.com

Search Github

# ventrillo_srv.ini
# Ventrilo Configuration

- File

org:company filename:ventrillo_srv.ini

org:company AdminPassword AND Password

user:name filename:ventrillo_srv.ini

user:name AdminPassword AND Password

"company.com" AdminPassword AND Password

github.com

Search Github

Slack Token : (xox[p|b|o|a]-[0-9]{12}-[0-9]{12}-[0-9]{12}-[a-z0-9]{32})

Slack Webhook : https://hooks.slack.com/services/T[a-zA-Z0-9_]{8}/B[a-zA-Z0-9_]{8}/[a-zA-Z0-9_]{24}

curl -sX POST "https://slack.com/api/auth.test?token=xoxp-TOKEN_HERE&pretty=1"

- Writeup
- Resource

org:company token=xoxp

org:company xoxp

user:name token=xoxp

user:name xoxp

"company.com" xoxp

Search Github

**Facebook Access Token : EAACEdEose0cBA[0-9A-Za-z]+**

**Facebook OAuth : [f|F][a|A][c|C][e|E][b|B][o|O][o|O][k|K].{0,30}['\"\\s][0-9a-f]{32}['\"\\s]**

**curl -s "https://developers.facebook.com/tools/debug/accesstoken/?access_token=ACCESS_TOKEN"**

- **Resource**

**org:company access_token=EAACEdEose0cBA**

**org:company EAACEdEose0cBA**

**user:name access_token=EAACEdEose0cBA**

**user:name EAACEdEose0cBA**

**"company.com" EAACEdEose0cBA**

github.com

Search Github

Google Api Key : AIza[0-9A-Za-z-_]{35}
Google OAuth : ya29\\.[0-9A-Za-z\\-_]+
Google OAuth ID : [0-9(+-[0-9A-Za-z_]{32}.apps.googleusercontent.com
Google Captcha : 6L[0-9A-Za-z-_]{38}
https://maps.googleapis.com/maps/api/staticmap?center=45%2C10&zoom=7&size=400x400&key=KEY

- Resource

org:company AIza

org:company key=AIza

org:company ya29

org:company secret=6L

Search Github

# Amazon AWS Access Key ID : AKIA[0-9A-Z]{16}

- **Resource**

```
org:company AWS_ACCESS_KEY_ID=AKIA
org:company AWSAccessKeyId=A3T
org:company AWS_ACCESS_KEY_ID=AGPA
org:company AWSAccessKeyId=ANPA
org:company AWS_ACCESS_KEY_ID=ASIA
"company.com" MWSAuthToken=amzn.mws
```

github.com

Search Github

## Twitter OAuth

[t|T][w|W][i|I][t|T][t|T][e|E][r|R].{0,30}['\"\\s][0-9a-zA-Z]{35,44}['\"\\s]

curl -u 'API key:API secret key' --data 'grant_type=client_credentials' 'https://api.twitter.com/oauth2/token'

curl --request GET --url https://api.twitter.com/1.1/account_activity/all/subscriptions/count.json --header 'authorization: Bearer TOKEN'

- Resource

org:company "API key"

org:company "authorization: Bearer"

user:name "API key"

user:name "authorization: Bearer"

github.com

Search Github

**Square OAuth Secret : sq0csp-[ 0-9A-Za-z\\-_]{43}**
**Square Access Token : sq0atp-[0-9A-Za-z\\-_]{22}**
**Auth Token :  EAAA[a-zA-Z0-9]{60}**

**curl https://connect.squareup.com/v2/locations -H "Authorization: Bearer [AUHT_TOKEN]"**

- **Resource**

  - org:company authorization AND Client AND sq0
  - org:company access_token:sq0atp
  - org:company authorization AND Client AND EAAA
  - org:company client_id:sq0

github.com

Search Github

# Mailgun API : key-[0-9a-zA-Z]{32}

curl --user 'api:key-PRIVATEKEYHERE' "https://api.mailgun.net/v3/domains"

- Resource

org:company "api:key-"

org:company mailgun AND key

user:name "api:key-"

"company.com" api:key

github.com

Search Github

Stripe standard api : sk_live_[0-9a-zA-Z]{24}

Stripe restricted api : rk_live_[0-9a-zA-Z]{24}

curl https://api.stripe.com/v1/charges -u TOKEN-HERE

- Resource

org:company sk_live_

org:company rk_live_

user:name sk_live_

user:name rk_live_

"company.com" stripe AND sk_live_ OR rk_live_

Search Github

# Heroku API

[h|H][e|E][r|R][o|O][k|K][u|U].{0,30}[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}

curl -X POST https://api.heroku.com/apps
-H "Accept: application/vnd.heroku+json; version=3" -H "Authorization: Bearer API_KEY_HERE"

- Resource

org:company Authorization AND Heroku

user:name Authorization AND Heroku

"company.com" Authorization AND Heroku

github.com

Search Github

https://api.hubapi.com/owners/v2/owners?hapikey=keyhere

https://api.hubapi.com/contacts/v1/lists/all/contacts/all?hapikey=keyhere

- Resource

org:company hapikey

org:company hubapi AND hapikey

user:name hapikey

user:name hubapi AND hapikey

"company.com" Authorization OR Token AND hapikey

github.com

Search Github

curl -s https://app.pendo.io/api/v1/feature
-H 'content-type: application/json' -H 'x-pendo-integration-key:KEY'
curl https://app.pendo.io/api/v1/metadata/schema/account -H 'content-type: application/json'

- Resource

org:company x-pendo-integration-key

org:company x-pendo-key

user:name x-pendo-integration-key

user:name x-pendo-key

"company.com"  x-pendo-key AND x-pendo-integration-key

Search Github

curl -H "x-api-key: APIKEYHERE" "https://console.jumpcloud.com/api/systems

- Writeup
- Resource

org:company x-api-key

org:company req.Header.Add AND key

user:name req.Header.Add AND key

user:name x-api-key

"company.com" jumpcloud AND x-api-key

github.com

Search Github

# Mapbox Secret Keys Rest Start With
# pk public token - sk secret token - tk temporary token

curl "https://api.mapbox.com/geocoding/v5/mapbox.places/Los%20Angeles.json?access_token=TOKEN"

- Resource

org:company access_token=sk

org:company access_token AND sk.eyJ

user:name access_token=sk

user:name access_token AND sk.eyJ

"company.com" mapbox AND access_token OR sk.eyJ

github.com

Search Github

```
curl --request PUT --url
https://<application-id>-1.algolianet.com/1/indexes/<example-index>/settings
-H 'content-type: application/json' -H 'x-algolia-api-key: <example-key>'
-H 'x-algolia-application-id: <example-application-id>'
--data '{"highlightPreTag": "<script>alert(1);</script>"}'
```

- Resource

org:company x-algolia-api-key

org:company algolia AND key

user:name x-algolia-api-key

user:name algolia AND key

"company.com" algolia AND x-algolia-api-key

github.com

Search Github

curl -X GET "https://api.pagerduty.com/schedules" -H "Authorization: Token token=TOKEN" -H "Accept: application/vnd.pagerduty+json;version=2"

- Resource

org:company pagerduty AND authorization AND token

org:company pagerduty AND token

user:name pagerduty AND authorization AND token

user:name pagerduty AND token

"company.com" pagerduty AND token

Search Github

curl "https://api2.branch.io/v1/app/KEY_HERE?branch_secret=SECRET_HERE"

- **Resource**

org:company branch_secret=

org:company branch AND branch_secret

user:name branch_secret=

user:name branch AND branch_secret

"company.com" branch AND branch_secret

Search Github

curl
"https://api.wpengine.com/1.2/?method=site&account_name=&wpe_apikey="

- **Resource**

org:company wpe_apikey=

org:company wpengine AND wpe_apikey

user:name wpe_apikey=

user:name wpengine AND branch_secret

"company.com" wpengine AND wpe_apikey

Search Github

curl "https://api.datadoghq.com/api/v1/dashboard?api_key=&application_key="

- **Resource**

org:company api_key AND application_key

org:company datadoghq AND api_key

user:name api_key AND application_key

user:name datadoghq AND api_key

"company.com" api_key AND application_key

Search Github

curl "https://gitlab.example.com/api/v4/projects?private_token=access_token"

- **Resource**

org:company private_token

org:company gitlab AND private_token

user:name private_token

user:name gitlab AND private_token

"company.com" gitlab AND private_token

github.com

Search Github

# curl "https://circleci.com/api/v1.1/me?circle-token=TOKEN"

- Writeup

org:company circle-token

org:company circle AND token

user:name circle-token

user:name circle AND token

"company.com" circle AND token

github.com

Search Github

# curl u USERNAME:KEY "https://REDACTED/artifactory/api/build"

- **Writeup**

org:company artifactory_key

org:company artifactory AND key

user:name artifactory_key

user:name artifactory AND key

"company.com" artifactory AND key

github.com

Search Github

# JSON Web Token
# Base64({}).Base64({}).Base64()
# eyJ---------.eyJ----------.-----------

- Mine

org:company .eyJ

org:company Authorization OR JWT AND eyJ

user:name .eyJ

user:name Authorization OR JWT AND eyJ

"company.com" Authorization OR JWT AND eyJ

# General Queries

Search Github

"Company" security_credentials

"Company" connectionstring

"Company" JDBC

"Company" ssh2_auth_password

"Company" ssh2_auth_password NOT string

"Company" send_keys OR sendkeys

"Company" JIRA_Pass

"Company" consumerkey

"Company" password OR pwd

"Company" auth_key

"Company" SSO_LOGIN

"Company" secret_access_key

"Company" bucket_password

"Company" redis_password

"Company" root_password

"Company" _TOKEN OR _KEY

attacker

automation approach

# Github-employees.py

## Find GitHub Accounts of Employees Of A Company Through Google Search

```
root@mine:~#puthon3 github-employees.py -m linkedin -t "company" -p 3
root@mine:~#puthon3 github-employees.py -f "facebook cookie" -o "github token" -t "company"
```

-m module to use to search employees on google , available linkedin AND github

-t term to search usually company name

-p number of  page to grab

" -f facebook Cookie " OR set VAR env FACEBOOK_COOKIE

" -o  github token " OR create a file called .token in the same directory with one token per line

attacker

automation approach

# Gitrob

## Find Potentially Sensitive Files Pushed To Public Repositories On Github

```
root@mine:~#./gitrob ORG-name
```

" -save ~/ORG-session.json " save the session to a file

" -load ~/ORG-session.json " load session stored in a file ORG-session.json

" -github-access-token github token " OR set VAR env GITROB_ACCESS_TOKEN

" -bind-address AND -port " Address to bind web server , default http://127.0.0.1:9393

attacker

automation approach

# Github-grabrepo.php

## Clones All Public Repositories Belonging To User OR organization

```
root@mine:~#php github-grabrepo.php -o ORG-name -d PATH-to-save
```

" -o ORG-name" clone all public repositories from ORG-name

" -u USER-name" clone all public repositories from USER-name

" -d PATH-to-save " dir to save all public repositories from ORG-name OR USER-name

attacker

automation approach

# Gitleaks

## SAST Tool For Detecting Hardcoded Secrets Remote Git Repos OR Local Git Repos

```
root@mine:~#./gitleaks --repo=https://github.com/org-OR-user/repo.git
root@mine:~#./gitleaks --repo-path=path-to-local-repo
```

" --repo=https://github.com/org-OR-user/repo.git " scan on a remote repo

" --repo-path=path-to-local-repo " scan on a local repo

" --verbose --pretty " which will output information in nicer looking json format output

" --org=ORG-name " organization to scan

" --user=USER-name " user to scan

attacker

automation approach

# truffleHog

## Gigging Deep Into Commit History AND Branches

```
root@mine:~#trufflehog --regex --entropy=False https://github.com/org-OR-user/repo.git
root@mine:~#trufflehog path-to-local-repo
```

" https://github.com/org-OR-user/repo.git " scan on a remote repo

" path-to-local-repo " scan on a local repo

--regex enable high signal regex checks

--json json format output

attacker

automation approach

# Github-dorks.py

## Performs Dorks On GitHub For The Users AND organizations

```
root@mine:~#python3 github-dorks.py -t GITHUB-token -o ORG-name -d PATH-to-dorks-file
```

" -t GITHUB-token " OR create a file called .token in the same directory with one token per line

" -d PATH-to-dorks-file " dorks file e.g. filename:sshd_config

" -o ORG-name " organization name to scan

" -u USER-name " user name to scan

attacker

automation approach

# GitDorker

## Utilizes The GitHub Search API To Provide Sensitive Information Stored On Github

```
root@mine:~#python3 gitdorker.py -t GITHUB-token -q QUERY -d PATH-to-dorks-file
```

" -t GITHUB-token " OR " -tf  FILE-with-tokens " file with one token per line

" -d PATH-to-dorks-file " dorks file e.g. filename:sshd_config

" -org ORG-name " organization name to scan

" -u USER-name " user name to scan

" -o PATH-to-save-output " path to save output

" -q QUERY " query search with list of dorks

attacker

automation approach

# gitGraber

## Monitor GitHub To Search AND Find Sensitive Data In Real Time

```
root@mine:~#python3 gitgraber.py -q \"ORG.com\" -k PATH-to-dorks-file -s -m
```

" -q \"ORG.com\" " query search with list of dorks

" -k PATH-to-dorks-file " dorks file e.g. filename:sshd_config

-s enable slack notifications

-m enable monitoring of your search every 30 m

# Thank You

**Mahmoud M. Awali**

@0xAwali