What is TruffleHog?

TruffleHog is a Python script with a dependency on the GitPython module that is designed to find potential secrets with a git repository by using entropy analysis. It won’t just find secrets that exist in the current code base, but will also check the repository's history for secrets so you can see secrets that may have been previously exposed. Since it only works on the git repository history, and not just the current code, it is better used as the first tool for finding secrets.

Installation of TruffleHog Using Pip:

Installation steps for pip:

1. for updating packages

sudo apt update

2. to install pip for python3

sudo apt install python3-pip

3. to verify installation of pip

pip3 –version

Installation of TruffleHog:

pip3 intsall truffleHog

How TruffleHog works?

This module will go through the entire commit history of each branch, and check each diff from each commit, and check for secrets. This is both by regex and by entropy. For entropy checks, truffleHog will evaluate the shannon entropy for both the base64 char set and hexidecimal char set for every blob of text greater than 20 characters comprised of those character sets in each diff. If at any point a high entropy string >20 characters is detected, it will print to the screen.

Usage of TruffleHog:

trufflehog [-h] [--json] [--regex] [--rules RULES]

[--entropy DO\_ENTROPY] [--since\_commit SINCE\_COMMIT]

[--max\_depth MAX\_DEPTH]

git\_url

Find secrets hidden in the depths of git.

positional arguments:

git\_url URL for secret searching

optional arguments:

-h, --help show this help message and exit

--json Output in JSON

--regex Enable high signal regex checks

--rules RULES Ignore default regexes and source from json list file

--entropy DO\_ENTROPY Enable entropy checks

--since\_commit SINCE\_COMMIT

Only scan from a given commit hash

--max\_depth MAX\_DEPTH

The max commit depth to go back when searching for

secrets

-i INCLUDE\_PATHS\_FILE, --include\_paths INCLUDE\_PATHS\_FILE

File with regular expressions (one per line), at least

one of which must match a Git object path in order for

it to be scanned; lines starting with "#" are treated

as comments and are ignored. If empty or not provided

(default), all Git object paths are included unless

otherwise excluded via the --exclude\_paths option.

-x EXCLUDE\_PATHS\_FILE, --exclude\_paths EXCLUDE\_PATHS\_FILE

File with regular expressions (one per line), none of

which may match a Git object path in order for it to

be scanned; lines starting with "#" are treated as

comments and are ignored. If empty or not provided

(default), no Git object paths are excluded unless

effectively excluded via the --include\_paths option.

What’s new in TruffleHog?

Searches through git repositories for secrets, digging deep into commit history and branches. This is effective at finding secrets accidentally committed.

truffleHog previously functioned by running entropy checks on git diffs. This functionality still exists, but high signal regex checks have been added, and the ability to surpress entropy checking has also been added.

Now that you have installed truffleHog, run it against a git repo. The command takes a github path, which means you can scan either a local or remote repository as follows:

Example for running truffleHog against a Bitbucket repository:

truffleHog --regex --entropy=False  <https://imbhascker13@bitbucket.org/accelerators-devops/bike-showroom-app.git>

Advantages of TruffleHog:

truffleHog’s output is more friendly than git-secrets’ and works by default, scanning the Git history. On the downside, its matching is not as configurable as git-secrets, and it lacks the ability to ignore files or commits and does not have the ability to check commit ranges.

On the other hand, truffleHog, will miss access keys but not secret keys