# TruffleHog Documentation

## Introduction

TruffleHog is a robust security tool designed to detect and analyze sensitive information, such as API keys, passwords, and private keys, within various data sources, including Git repositories, filesystems, and cloud storage. It aids in identifying potential security risks by scanning for exposed secrets across different platforms.

## Key Features

- Discovery: TruffleHog can search for secrets in multiple locations, including Git repositories, chats, wikis, logs, API testing platforms, object stores, and filesystems.  
- Classification: It classifies over 800 types of secrets, mapping them back to their specific identities, such as AWS secrets, Stripe secrets, Cloudflare secrets, Postgres passwords, and SSL private keys.  
- Validation: TruffleHog includes built-in validation logic for certain secret types, confirming their validity and activity.

## Installation

TruffleHog can be installed using various methods, including package managers like APT and Homebrew, Docker, binary releases, or by cloning and compiling from the source. Detailed installation instructions are available in the [TruffleHog GitHub repository](https://github.com/trufflesecurity/trufflehog#floppy\_disk-installation).

## Usage

TruffleHog operates through several subcommands, each tailored to scan different data sources:

- git: Scans Git repositories.  
- github: Scans GitHub repositories.  
- gitlab: Scans GitLab repositories.  
- docker: Scans Docker images.  
- s3: Scans Amazon S3 buckets.  
- filesystem: Scans local files and directories.  
- syslog: Scans syslog data.  
- circleci: Scans CircleCI configurations.  
- gcs: Scans Google Cloud Storage buckets.

Each subcommand has specific options and flags to customize the scanning process. For example, to scan a Git repository, you can use:

trufflehog git [options] <uri>

To view the available options for a subcommand, use the `--help` flag:

trufflehog git --help

## Common Flags and Options

- `--branch <branch>`: Specifies the branch to scan in a Git repository.  
- `--since-commit <commit>`: Scans all changes in a new branch not yet merged into the main branch.  
- `--json` or `--json -o <file>`: Outputs the results in JSON format, optionally writing to a specified file.  
- `--entropy` or `--no-entropy`: Enables or disables entropy-based secret detection.  
- `--regex` or `--no-regex`: Enables or disables regex-based secret detection.  
- `--exclude-paths <regex>`: Excludes paths matching the specified regular expression from the scan.  
- `--config <file>`: Specifies a configuration file to use for the scan.

## Best Practices

- Regular Scanning: Integrate TruffleHog into your CI/CD pipelines to ensure continuous monitoring for exposed secrets.  
- Custom Configuration: Utilize custom regex patterns and entropy thresholds to fine-tune the scanning process and reduce false positives.  
- Secret Management: Implement robust secret management practices, such as using environment variables or secret management tools, to prevent hardcoding secrets in your codebase.