Hooks, Pre-commit, pre-push

Every time a particular event occurs within a repository, they let you customize its internal behavior and triggered customizable actions at key points in the development lifecycle.

So if you have any sensitive information like access keys, access tokens as a set of keys, these are
often lead to accidental good comments so precommitment can be installed on a workstation to avoid that.So this hoax usually work on a project based approach for filtering the sensitive data.
And if the developers want, they can still bypass the.

So you have various tools in the market which helps us with pre commitment push.
So one of the tools is Talisman, which is an open source tool created by Thought works.
So Talisman is an open source tool which can be used to hook your repository to ensure that potential
secret or sensitive information does not leave the developers workstations.
It validates the outgoing change for things that look suspicious, like a potential password, tokens,
private keys, etc..
It can also be used as a repository history scanner to detect secrets that have already been checked
in so that you can take an informed decision to safeguard the secrets.

Download the talisman installer script

```
curl https://thoughtworks.github.io/talisman/install.sh> ~/install-talisman.sh
chmod +x ~/install-talisman.sh
```

Install to a single project

```
cd my-git-project
# as a pre-push hook
~/install-talisman.sh
# or as a pre-commit hook
~/install-talisman.sh pre-commit
```

Create sec_files

```
mkdir sec_files && cd sec_files
echo "username=sidd-harth" > file1
echo "secure-password123" > password.txt
echo "apikey=AizaSyCqhjgrPtr_La56sdUkjfav_laCqhjgrPtr_2s" > file2
echo "base64encodedsecret=cGFzc3dvcmQtaXMtcXdlcnR5MTIzCg==" > file3
```

Commit and see report.

```
git commit -m "add"
Talisman Scan: 12 / 12 <----------------------------------------------------------> 100.00%

Talisman Report:
+----------------------+----------------------------------------------------+----------+
|        FILE          |                    ERRORS                          | SEVERITY |
+----------------------+----------------------------------------------------+----------+
| sec_files/password.txt | The file name                                    | low      |
|                      | "sec_files/password.txt"                           |          |
|                      | failed checks against the                          |          |
|                      | pattern password                                   |          |
+----------------------+----------------------------------------------------+----------+
| sec_files/file3      | Expected file to not to contain                    | high     |
|                      | base64 encoded texts such as:                      |          |
|                      | base64encodedsecret=cGFzc3dvcmQtaXMtcXdlcnR5MTI... |          |
+----------------------+----------------------------------------------------+----------+
| sec_files/file3      | Potential secret pattern :                         | low      |
|                      | base64encodedsecret=cGFzc3dvcmQtaXMtcXdlcnR5MTIzCg== |        |
+----------------------+----------------------------------------------------+----------+
| sec_files/file2      | Potential secret pattern :                         | low      |
|                      | apikey=AizaSyCqhjgrPtr_La56sdUkjfav_laCqhjgrPtr_2s |         |
+----------------------+----------------------------------------------------+----------+
```

If you are absolutely sure that you want to ignore the above files from talisman detectors, consider pasting the following format in .talismanrc file in the project root

```
fileignoreconfig:
- filename: sec_files/password.txt
  checksum: f1cfc4e74637c1399a305acaf7991724f54503ec159d4f6ad969cacdfb8e04c8
- filename: sec_files/file3
  checksum: b058bbb84595454d550863a6ae7fd501863acd9692ec3cc699bc6f76dd0e38a5
- filename: sec_files/file2
  checksum: b39b1a7ab78830bc35571285a210d1d0b0aa2b213caf2fea02cee664c473b237
version: ""

Talisman done in 111.668199ms
root@master-node:/home/dai/devsecops# git push
Username for ' https://github.com': hiep98
Password for ' https://hiep98@github.com':
Talisman Scan: 0 <.........................................................> ?%
Talisman done in 29.34035ms
Everything up-to-date
```

Igorn:
```
Talisman done in 69.02484ms
cat .talismanrc
fileignoreconfig:
```

```
- filename: sec_files/password.txt
  checksum: f1cfc4e74637c1399a305acaf7991724f54503ec159d4f6ad969cacdfb8e04c8
- filename: sec_files/file3
  checksum: b058bbb84595454d550863a6ae7fd501863acd9692ec3cc699bc6f76dd0e38a5
```

Commit again

```
git add .
git commit -m "addd2"
Talisman Scan: 15 / 15 <-------------------------------------------------------> 100.00%
Talisman Report:
+----------------+---------------------------------------------------+----------+
|     FILE       |                   ERRORS                 | SEVERITY |
+----------------+---------------------------------------------------+----------+
| sec_files/file2 | Potential secret pattern :                 | low      |
|                | apikey=AizaSyCqhjgrPtr_La56sdUkjfav_laCqhjgrPtr_2s |          |
+----------------+---------------------------------------------------+----------+
```

If you are absolutely sure that you want to ignore the above files from talisman detectors, consider pasting the following format in .talismanrc file in the project root

```
fileignoreconfig:
- filename: sec_files/file2
  checksum: b39b1a7ab78830bc35571285a210d1d0b0aa2b213caf2fea02cee664c473b237
version: ""
```