GitGuardian [Rewriting git history cheat sheet — 2020]

HOW TO USE FLOWCHART

01. Select a scenario that applies at Step 1 — **02.** In all cases move to Step 2 — **03.** Move to Step 3 that relates to the original scenario

1 — 2 — 3

SCENARIO A

The secret is only commited locally

If you have uncommited work:

```
git stash
```

SCENARIO B

The secret has been pushed to a remote repository but **no team members working on effected branch**

If you have uncommited work:

```
git stash
```

SCENARIO C

The secret has been pushed to a remote repository with **team members on the same branch (this can get complicated)**

If you have uncommited work:

```
git stash
```

Notify any team members working on the same branch(es). Have them commit and push up all work. Close any outstanding PRs on the branch(es). Then:

```
git clone [URL] && cd
[project_name]
&& git pull --all --tags
```

ALL SCENARIOS

Which is applicable?

Secret in last commit:

```
git reset --hard HEAD~1
```

→ 3

Secret in last commit with other unrelated changes:

```
git add [FILENAME]
git commit --amend
```

→ 3

Secret is not in last commit therefore it's complicated. install git-filter-repo

Is the secret a portion of a larger file or a file that needs to be removed?

Portion of a file:
Create file «replacement.txt»
Use format:
ORIGINAL==>REPLACEMENT

```
git filter-repo
--replace-text
../replacements.txt
[--force]
```

Entire file:

```
git filter-repo
--use-base-name --path
[FILENAME]
--invert-paths
```

→ 3

SCENARIO A

```
git stash pop
```

→ ✓

SCENARIO B

```
git push --all --tags
--force
```

↓

```
git stash pop
```

→ ✓

SCENARIO C

```
git push --all --tags
--force
```

↓

```
git stash pop
```

↓

Did teammates get their work up before you made the clone?

YES → ✓

NO ↓

Teammates need to rebase their outstanding work.

```
git fetch
git rebase -i
origin/[branchname]
```

→ ✓