

Change Commit history file

Hãy cẩn thận khi thay đổi lịch sử commit luôn luôn phải backup trước khi làm theo hướng dẫn để phòng rủi ro trong quá trình thay đổi có thể gây ra lỗi

Trong quá trình code thì việc các lập trình viên có xu hướng hardcode những thông tin quan trọng như **Passwords, Credentials & Private data** và sau đó commit những thông tin đó lên lên gitlab. Dù sau đó có được phát hiện và sửa thì những commit history vẫn lưu trữ những thông tin nhạy cảm. Rủi ro nếu branch của repo bị lộ ra thì hacker có khả năng sẽ lấy được những thông tin đó và dùng những thông tin đó để khai thác lỗ hổng thật là lạy ông tôi ở bụi này. Vì vậy dưới đây là bài viết hướng dẫn thay đổi các commit được lưu trữ trong lịch sử gitlab nhằm đảm bảo nếu có bị lộ branch repo thì những thông tin nhạy cảm vẫn được đảm bảo bí mật

Git command

Git command cung cấp một Command là git-filter-branch dùng để trouble shooting các vấn đề về branch commit v.v.

```
git filter-branch -f --tree-filter "find . -type f -exec sed -i '' -e 's/  
originalpass/newpass/g' {} \;"  
#command trên sẽ trực tiếp thay đổi vào commit history nên hãy cân nhắc backup  
kỹ càng trước khi push lên  
git push -f
```

BFG Repo-Cleaner

Đây là một công cụ hỗ trợ một giải pháp tương tự như git-filter-branch tuy không mạnh mẽ và làm được nhiều việc như git-filter-branch nhưng *bfg* làm tốt những công việc mà chúng ta cần như viết lại lịch sử commit hay xóa những file tệp quan trọng mà ta commit lên

Để cài đặt thì ta cần phải có homebrew hoặc phải cài java để chạy

Có 2 cách cài đặt

Cài thông qua homebrew

```
brew install bfg  
bfg  
bfg 1.13.0
```

```
Usage: bfg [options] [<repo>]
```

```
-b, --strip-blobs-bigger-than <size>
                                strip blobs bigger than X (eg '128K', '1M', etc)
-B, --strip-biggest-blobs NUM
                                strip the top NUM biggest blobs
-bi, --strip-blobs-with-ids <blob-ids-file>
                                strip blobs with the specified Git object ids
```

Hoặc tải file jar về để chạy

Link tải : <https://repo1.maven.org/maven2/com/madgag/bfg/1.13.0/bfg-1.13.0.jar>

```
java -jar bfg.jar
bfg 1.13.0
Usage: bfg [options] [<repo>]

-b, --strip-blobs-bigger-than <size>
                                strip blobs bigger than X (eg '128K', '1M', etc)
-B, --strip-biggest-blobs NUM
                                strip the top NUM biggest blobs
-bi, --strip-blobs-with-ids <blob-ids-file>
                                strip blobs with the specified Git object ids
```

Đầu tiên tạo file replace.txt và thêm vào file giá trị muốn thay đổi theo format sau

```
regex:<VALUE>==>'*****'
example:
#giá trị muốn ẩn là 123123123 thì trong file replace.txt sẽ là
regex:123123123==>'*****'
#mục đích là thay đổi tất cả giá trị 123123123 thành *****
```

Sau đó ta chạy lệnh như sau để tiến hành

```
bfg --replace-text "replace.txt"
#mặc định không thêm repo phía sau thì bfg tự nhận thư mục hiện tại là repo,
```

```
đọc out oput thật kỹ và kiểm tra nếu mọi thứ ổn thì chạy lệnh dưới
git reflog expire --expire=now --all && git gc --prune=now --aggressive
git add .
git commit -m "*****"
git push -f #phải có flag -f để thay đổi toàn bộ lịch sử nếu không sẽ bị lỗi
```

Dưới đây là một script sample để tham khảo

change_commit.sh

Đây là script sẽ thay những thông tin nhạy cảm thành một chuỗi "" được lưu trữ ở commit history mà không phải xóa các commit đi nhằm hạn chế ảnh hưởng nếu như những commit đó có sửa bug hay những đoạn code quan trọng cho những người sau có thể đọc và hiểu được.

| Viết lại lịch sử thật dễ dàng với *change_commit.sh*

Script này yêu cầu phải có **brew** để có thể chạy được nhưng hãy yên tâm nếu bạn chưa có thì script này cũng hỗ trợ cài đặt cho bạn mà bạn không cần phải lo chưa phải tận thế đâu

Bây giờ ta sẽ chạy File script bằng hai cách

1.

```
chmod +x change_commit.sh
./change_commit.sh
```

Với cách một thì chỉ cần chạy lệnh chmod lần đầu tiên chạy còn những lần sau chỉ cần thực thi lệnh thứ 2

2.

```
bash change_commit.sh
```

Nếu chưa cài đặt những phần mềm cần thiết thì script sẽ cài đặt chúng và đây là giao diện sau khi đã cài đặt đầy đủ

Ta cần phải nhập đường dẫn thư mục bắt đầu từ root như hình dưới thông thường cách

đơn giản nhất là vào thư mục project nhấn lệnh `pwd` sau đó copy rồi dán vào đây như hình dưới là được

```
→ change_history_commit git:(master) bash change_commit.sh
brew is installed
bfg is installed
=====
Input folder path
Must be from root folder patch, use pwd command at the folder of project :
/Users/khoint/project/testcommit
```

Sau khi thư mục được kiểm tra là tồn tại thì đoạn dưới sẽ hiện ra

```
Check the git remote to make sure you are at the right repo:
```

```
origin https://github.com/khoint0210/testcommit.git (fetch)
origin https://github.com/khoint0210/testcommit.git (push)
Is the repo correct (y/n)
```

Sẽ đưa ra remote repo để kiểm tra xem có đúng repo chúng ta muốn sửa commit history hay không chọn y hoặc yes , n hoặc no để tiếp tục

Nếu chọn y thì mọi việc tới giờ vẫn ổn thỏa và hình như bên dưới sẽ hiện ra để sử dụng.

```
Making backup folder
which value you want to hide
```

Đồng thời script cũng đã tạo ra một file backup của thư mục repo nhằm trong trường hợp script lỗi hoặc thao tác sai theo format bên dưới

```

→ project ls -t testcommit-backup-*
testcommit-backup-08_13_15_42:
replace.txt test.txt      secret.txt

testcommit-backup-08_13_20_14 7:
replace.txt test.txt      secret.txt

testcommit-backup-08_13_20_1358:
replace.txt test.txt      secret.txt

```

<tên thư mục>-backup-<ngày-tháng-giờ-phút>

Và bản sao lưu này sẽ nằm cùng thư mục với thư mục project mà ta đang sử dụng

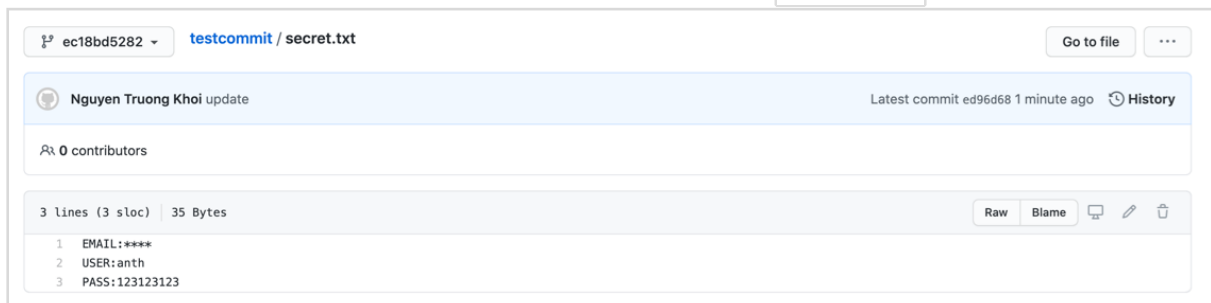
```

→ change_history_commit git:(master) bash change_commit.sh
brew is installed
bfg is installed
=====
Input folder path
Must be from root folder patch, use pwd command at the folder of project :
/Users/khoiint/project/testcommit
Check the git remote to make sure you are at the right repo:

origin https://github.com/khoiint0210/testcommit.git (fetch)
origin https://github.com/khoiint0210/testcommit.git (push)
Is the repo correct (y/n)
y
Making backup folder
which value you want to hide
█

```

Bây giờ là lúc “viết lại lịch sử” giả sử thông tin nhạy cảm là 123123123



thì ta sẽ nhập vào và thông tin nhạy cảm là 123123123 sau đó nhấn enter

```

...Ref update completed in 32 ms.

Commit Tree-Dirt History
-----

      Earliest                      Latest
      |                             |
      .....DDm

D = dirty commits (file tree fixed)
m = modified commits (commit message or parents changed)
. = clean commits (no changes to file tree)

-----
      Before      After
-----
First modified commit | ed96d687 | c77d953e
Last dirty commit     | ec18bd52 | 479b70be

Changed files
-----

      Filename      Before & After
-----
secret.txt | 6f60ca45 → 317c426a

In total, 5 object ids were changed. Full details are logged here:

/Users/khoaint/project/testcommit.bfg-report/2020-08-13/16-08-41

BFG run is complete! When ready, run: git reflog expire --expire=now --all && git gc --prune=now --aggressive

--
You can rewrite history in Git - don't let Trump do it for real!
Trump's administration has lied consistently, to make people give up on ever
being told the truth. Don't give up: https://github.com/bkeepers/stop-trump
--

check file carefully before process and push to git
Is it work correctly (y/n)

```

Ta có thể thấy commit cũ mà ta đối chiếu đã đổi từ `ed96d687` thành `c77d953e` và thấy được file có chứa thông tin nhạy cảm đã sửa là `secret.txt` đúng thì ta chọn “y” để script sẽ lưu lại và đẩy lên gitlab thay toàn bộ lịch sử commit

Sau khi push lên thành công

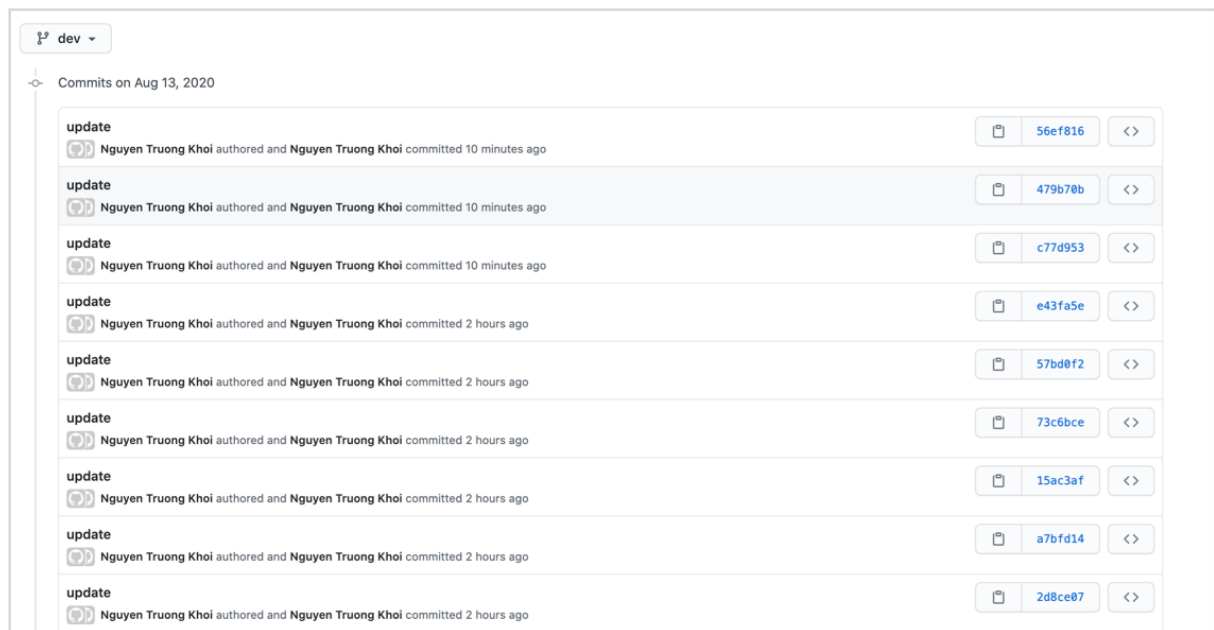
```

nothing to commit, working tree clean
do you want to push the local git
y
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (25/25), 2.10 KiB | 2.10 MiB/s, done.
Total 25 (delta 6), reused 25 (delta 6)
remote: Resolving deltas: 100% (6/6), done.
To https://github.com/khoaint0210/testcommit.git
+ 3da1f25...56ef816 dev -> dev (forced update)
Enumerating objects: 90, done.
Counting objects: 100% (90/90), done.
Delta compression using up to 12 threads
Compressing objects: 100% (62/62), done.
Writing objects: 100% (90/90), done.
Total 90 (delta 22), reused 68 (delta 0)
Computing commit graph generation numbers: 100% (35/35), done.
On branch dev
Your branch is up to date with 'origin/dev'.

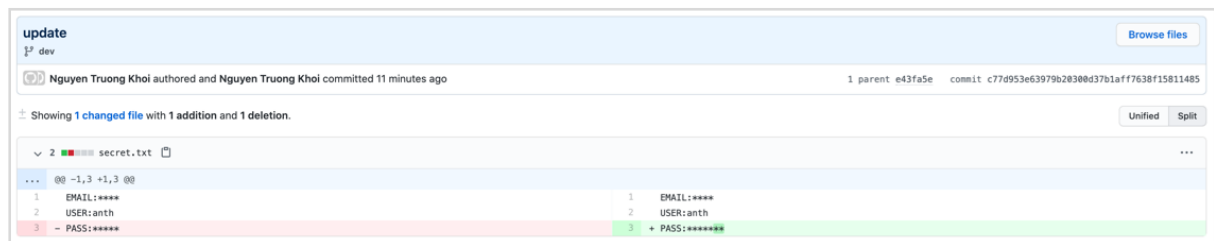
nothing to commit, working tree clean
Everything up-to-date

```

Ta xem lại lịch sử commit sẽ không thấy hash cũ nữa mà thay vào đó là hash mới



Và khi nhấn vào ta không còn thấy 123123123 nữa mà thay vào đó chỉ là



Các dấu mà không ai biết được thật sự là gì