Домашнее задание к занятию «1.3. Системы контроля версий и CICD

В качестве результата отправьте ответы на вопросы в личном кабинете студента на сайте netology.ru.

Предисловие

Данные ДЗ будут представлять собой лабораторные работы, в рамках которых вы по инструкциям выполните определённые шаги.

Задание truffleHog

Есть замечательный инструмент, который называется truffleHog. Он умеет искать по всей истории вашего проекта «секреты». Конечно же, если ваш проект использует git.

В этой лабораторной работе мы попробуем его в использовании на более-менее реалистичном примере, чтобы вы увидели, что не всё так гладко и достаточно часто бывает много ложных срабатываний, которые приходится «разгребать». Если вы просто отдадите «простыню» логов разработчикам, чтобы они сами разбирались — ничего хорошего из этого не выйдет.

Порядок выполнения

Для truffleHog нужен Python, устанавливается он с помощью Pip — системы управления пакетами для Python.

Вы можете проделать всё с помощью VM, но мы предлагаем рассмотреть возможность использования Docker Container' в качестве «одноразовой машины».

Итак, начнём:

1. Поскольку нам нужен Python, да ещё и с установленным менеджером пакетов Pip, то логично найти уже настроенный образ Python.

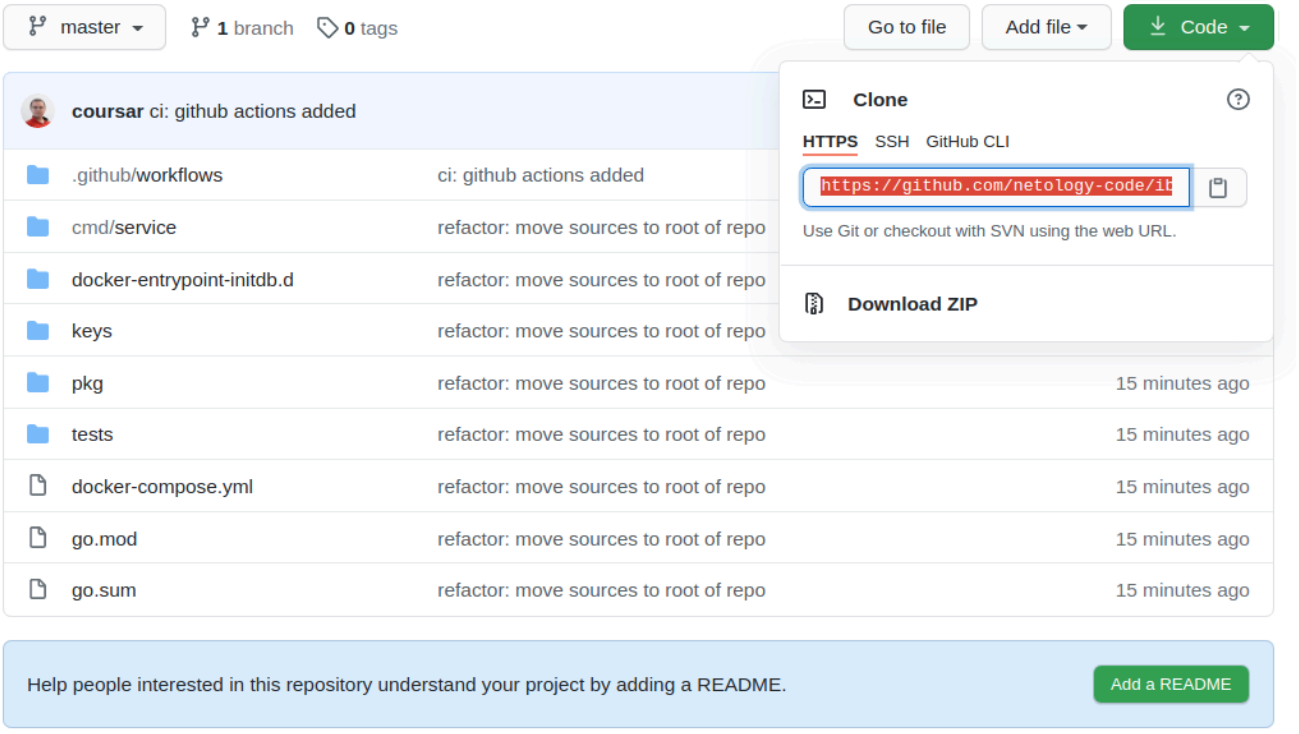
2. На момент написания этого ДЗ (2021 г.) в Docker Hub есть три ключевых группы образов: windowsservercore (нам не подходит), buster (на базе Debian Buster) и alpine (на базе Alpine). Мы будем использовать buster, поскольку он содержит уже привычный вам менеджер apt и всё остальное.

3. `docker run -it -p 8080:8080 python:buster /bin/bash`. Скоро узнаем зачем нам открывать порт.

4. После того как вы попали в терминал контейнера, необходимо установить truffleHog: `pip install truffleHog`. При этом обратите внимание на регистр.

5. Далее всё достаточно просто: указываем инструменту ссылку или путь к репозиторию. В нашем случае мы воспользуемся специально подготовленным: https://github.com/netology-code/ib-secrets.

Переходим по ссылке и кликаем на кнопке `Code`:



Важно: убедитесь, что вы выбрали именно HTTPS и скопировали ссылку.

6. Используйте следующую команду для запуска сканирования:

```
trufflehog https://github.com/netology-code/ib-secrets.git | tee -a log.txt
```

Обратите внимание на регистр. Да, это не опечатка, теперь `hog` написано маленькими буквами.

Вы увидите примерно следующий вывод:

```
~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~
Reason: High Entropy
Date: 2021-01-21 07:06:57
Hash: 357116ea59d9d0d4cfe8bed75c09da0f3ee99b2a
Filepath: service/go.sum
Branch: origin/master
Commit: feat(service): symmetric version added

@@ -1,182 +0,0 @@
-github.com/BurntSushi/toml v0.3.1/go.mod h1:xHWCNGjB5oqiDr8zfno3MHue2Ht5sIBksp03qcyfWMU=
-github.com/cockroachdb/apd v1.1.0/go.mod h1:8Sl8LxpKi29FqWXR16WEFZRNSz3SoPzUzeMeY4+DwBQ=
-github.com/coreos/go-systemd v0.0.0-20190321100706-95778dfbb74e/go.mod h1:F5haX7vjVVG0kc13fIWeqUViNPyEJxv/OmvnBo0Yme4=
-github.com/coreos/go-systemd v0.0.0-20190719114852-fd7a80b32e1f/go.mod h1:F5haX7vjVVG0kc13fIWeqUViNPyEJxv/OmvnBo0Yme4=
-github.com/creack/pty v1.1.7/go.mod h1:lj5s0c3V2DBrqTV7llrYr5NG6My20zk30Fl46Y7DoTY=
-github.com/creack/pty v1.1.9/go.mod h1:oKZEueFk5CKHvIhNR5MUki03XCEU+Q6VDXinZuGJ33E=
-github.com/davecgh/go-spew v1.1.0/go.mod h1:J7Y8YcW2NihsgmVo/mv3lAwl/skON4iLHjSsI+c5H38=
-github.com/davecgh/go-spew v1.1.1/go.mod h1:J7Y8YcW2NihsgmVo/mv3lAwl/skON4iLHjSsI+c5H38=
-github.com/go-chi/chi v1.5.0 h1:2ZcJZozJ+rj6BA0c19ffBUGXEKAT/aOLOtQjD46vBRA=
-github.com/go-chi/chi v1.5.0/go.mod h1:REp24E+25iKvxgeTfHmdUoL5x15kBiDBlnIl5bCwe2k=
```

Зелёным будет подсвечена метаинформация, а оранжевым — обнаруженный участок кода. В данном случае это ложные срабатывания, поскольку это явно не секреты, а checksum зависимостей нашего проекта.

Повторно запустить лог для просмотра вы можете с помощью команды `cat log.txt | more` (выход по клавише `q`).

7. Конечно, вы можете поставить с помощью `apt` или `apt update` какой-нибудь текстовый редактор и проанализировать `log.txt`. Но давайте посмотрим, как забрать файл из контейнера.

В Python есть встроенный модуль HTTP-сервера, который запускает HTTP-сервер в текущем рабочем каталоге: `python -m http.server 8080`:

P. S. В качестве дополнения поищите, как можно обойтись без Python с помощью `nc`.

Результаты выполнения

Отправьте в личном кабинете студента найденные вами секреты с описанием того, чтобы это могло быть по вашему мнению.

Дополнительно

Можете дополнительно ознакомиться со сравнением коммерческого сервиса GitGuardian с truffleHog.

Задание BFG Repo-Cleaner

Для удаления «чувствительных данных» существуют специальные инструменты. Самые простые:

1. BFG Repo-Cleaner.
2. Встроенная в `git` команда `filter-branch`.

Порядок выполнения

Разработчики уверяют, что вычистили все найденные вами в предыдущем задании секреты из истории с помощью BFG Repo-Cleaner. Убедитесь, так ли это.

Для этого используйте следующий репозиторий: https://github.com/netology-code/ib-secrets-fixed.git.

Результаты выполнения

Отправьте в личном кабинете студента общее заключение в свободной форме о выполнении разработчиками задачи по вычистке репозитория.