

Exercise 2

Tianfang Wang

a1788845

Vulnerabilities Assigned: CVE-2015-5174, CVE-2018-1325, CVE-2013-7251

Github Name: 1788845

Github Link: <https://github.com/1788845/SSES2>

For this exercise, I use Jupyter-notebook to analyze the Github's commits. All outputs are in the Exercise2_Tianfang_Wang_a1788845.ipynb.

Code details explanation:

```
repo1 = Repo("~/Documents/tomcat80")
fixing_commit_1 = "2fc9d03ffbc3fe7eabfd272380807ac0ddcf748d"
```

Import project and commit.

Q(a) What was the message and title of the mention of fixing a bug or vulnerability?

```
res1 = repo1.git.log(-1, '--pretty=%B', fixing_commit_1)
print("Title:", res1)
```

Command Git.log is used to get commit logs, “-1” is get the first row of the commit (Title). “Pretty=%B” is used for limiting the output format as raw body.

Q(b) How many total files were affected in the fixing commit?

```
res2 = repo1.git.diff('--name-only', fixing_commit_1, fixing_commit_1 + '^').splitlines()
print("Number of affected files: %d"%len(res2))
```

Command git.diff is used to show changes between commits. “—name-only” shows only names of changed files. “fixing_commit” is current commit, “fixing_commit + “^”” is previous commit. And use splitline() method to splits those names in to a list. Then output the length of this list is the result.

Q(c) How many total directories were affected in the fixing commit? For example, if a file path is: abc/def/File.java, then its directory is abc/def.

```
res3 = repo1.git.diff('--dirstat', fixing_commit_1, fixing_commit_1 + '^').splitlines()
print("Number of affected directories: %d"%len(res3))
```

Number of affected directories: 3

Similar to Q(b), “--dirstat” will count at level of directory.

Q(d) How many total lines of code (including comments and blank lines) were deleted?

```
res4 = repo1.git.show(fixing_commit_1).splitlines()
res4 = [line for line in res4 if re.compile('^-$|^-[-]').match(line)]
print("Number of deleted lines (including comments and blank lines): %d"%len(res4))
```

Number of deleted lines (including comments and blank lines): 21

Q(e) How many total lines of code (including comments and blank lines) were added?

```
res5 = repo1.git.show(fixing_commit_1).splitlines()
res5 = [line for line in res5 if re.compile('^\\+$|^\\+[\\+]').match(line)]
print("Number of added lines (including comments and blank lines): %d"%len(res5))
```

Number of added lines (including comments and blank lines): 98

Q(f) How many total lines of code (excluding comments and blank lines) were deleted?

```
res6 = repo1.git.show('--ignore-blank-lines', fixing_commit_1).splitlines()
res6 = [line for line in res6 if re.compile('^-[^-]').match(line)]
res6 = [line for line in res6 if not re.compile('^[-|]\\s*\\/').match(line)]
res6 = [line for line in res6 if not re.compile('^[-|]\\s*\\*').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res6))
```

Number of added lines (excluding comments and blank lines): 20

Q(g) How many total lines of code (excluding comments and blank lines) were added?

```
res7 = repo1.git.show('--ignore-blank-lines', fixing_commit_1).splitlines()
res7 = [line for line in res7 if re.compile('^\\+[\\+]').match(line)]
res7 = [line for line in res7 if not re.compile('^\\+[\\+]\\s*\\/').match(line)]
res7 = [line for line in res7 if not re.compile('^\\+[\\+]\\s*\\*').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res7))
```

Number of added lines (excluding comments and blank lines): 78

Q(d) to Q(g) are similar. I’m using regular expression to find the result. For example, “^-\$|^-[-]” This regular expression can be split as two parts. It means “^-\$” or “^-[^-]”. “^-\$” means start with “-” and end with “-”, which is deleted a blank line. “^-[^-]” means start with “-” and but not followed with “-”, which is excluding the scenario like “--”.

Q(h) How many days were between the current fixing commit and the previous commit of each affected file?

```
for file in repo1.git.diff('--name-only', fixing_commit_1, fixing_commit_1 + '^').splitlines():
    res8 = repo1.git.log(-2, '--pretty=%ct', file).splitlines();
    if len(res8) == 2:
        print(" For %s , the days between the current fixing commit and the previous commit is %.1f days"%(file, (int(res8[0])-int(res8[1]))/86400))
    else:
        print(" %s: N/A This file doesn't have previous commit"%file)
```

For java/org/apache/tomcat/util/http/RequestUtil.java , the days between the current fixing commit and the previous commit is 230.7 days
For test/org/apache/tomcat/util/http/TestRequestUtil.java , the days between the current fixing commit and the previous commit is 571.9 days
For webapps/docs/changelog.xml , the days between the current fixing commit and the previous commit is 26.1 days

For this question, first step is getting the file name, then use git.log to get 2 commit time. “—pretty=%ct” is limit the output in second. And then if we got 2 commit time, we calculate the different and then convert second to days, if we cannot get 2 commit time, it means this file is a new file which doesn't has previous commit.

Q(i) How many time has each affected file of the current fixing commit been modified in the past since their creation

```
for file in repo1.git.diff('--name-only', fixing_commit_1, fixing_commit_1 + '^').splitlines():
    res9 = repo1.git.log('--follow', '--pretty=oneline', file).splitlines();
    print(" %s has been modified %d times"%(file, len(res9)))
```

java/org/apache/tomcat/util/http/RequestUtil.java has been modified 6 times
test/org/apache/tomcat/util/http/TestRequestUtil.java has been modified 14 times
webapps/docs/changelog.xml has been modified 2919 times

Similar to Q(b), “—pretty=oneline” makes the output of history one in a row, and we can get the modified times by counting how many lines are there.

Q(j) Which developers have modified each affected file since its creation?

```
for file in repo1.git.diff('--name-only', fixing_commit_1, fixing_commit_1 + '^').splitlines():
    print(" %s has been modified by following developers:"%file)
    res10 = repo1.git.log('--follow', '--pretty=%aN', file).splitlines()
    for dev in set(res10):
        print(" %s"%dev)
```

java/org/apache/tomcat/util/http/RequestUtil.java has been modified by following developers:

Print file name then print all authors that modified this file. “--follow” will continue listing the history of a file beyond renames, and “—pretty=%aN” will list all authors.

Q(k) For each developer identified, how many commits have each of them submitted? From your observation, are the involving developers experienced (with many commits) or new ones (with few commits) or both?

```
: devs = []
for file in repo1.git.diff('--name-only',fixing_commit_1, fixing_commit_1 + '^').splitlines():
    devs += repo1.git.log('--follow', '--pretty=%aN', file).splitlines()
res11 = list(set(devs))
log = repo1.git.log('--pretty=%aN')
for dev in res11:
    print(" %s made %d commits"%(dev, len(re.findall(dev,log))))
```

Similar to Q(j), after getting all authors name, we count how many time their name appeared. For each appear, make the counter plus one, the final count result is how many commits they submitted.