

### Exercise 3

Tianfang Wang

a1788845

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

Github Name: 1788845

Github Link: <https://github.com/1788845/SSES3>

For this exercise, I use Jupyter-notebook to find out VCC and to analyze the Github's commits. VCC result is in Exercise3\_Tianfang\_Wang\_a1788845.ipynb, analysis result is in Exercise3\_Analysis\_Tianfang\_Wang\_a1788845.ipynb.

- Find VCC Code details explanation:

#### Loading modules

```
1 [40]: import os
import re
import numpy as np
# Specify git executable file for GitPython in Jupyter Notebook (In IDE, it can
os.environ["GIT_PYTHON_REFRESH"] = "quiet"
import git
from git import RemoteProgress
from git import Repo
```

First step is loading modules.

```
1]: def sumLines(dLines):
    pattern = re.compile('@@')
    return [s for s in dLines if pattern.match(s)]

def pSummary(summary):
    num = summary.split(' ')
    dNum = num[1].split(',')
    aNum = num[2].split(',')
    dStart = -int(dNum[0])
    if len(dNum)==1:
        dLength = 1
    else:
        dLength = int(dNum[1])
    aStart = int(aNum[0])
    if len(aNum)==1:
        aLength = 1
    else:
        aLength = int(aNum[1])
    return (dStart, dLength, aStart, aLength)
```

Then, find out the file changes by using command `git.diff('-U0',PREV,HEAD,file).splitlines()`.

It will show the result like “@@ -45,9, +45,10” This means there is a deletion start at line 45, total 9 lines deleted, and a addition start at line 45, total 10 lines. Collection those data, which can use later for finding enclosing cope.

```
def findScope(dStart, dLength, content):
    lineNum = dStart-1
    count = 1
    while count > 0:
        lineNum += 1
        if lineNum >= len(content):
            lineNum = len(content) - 1
            break
        line = content[lineNum]
        for c in line:
            if c=='}':
                count -= 1
                if count==0:
                    break
            if c=='{':
                count += 1
    sEnd = lineNum+1
    lineNum = dStart
    count = 1
    while count > 0:
        lineNum -= 1
        if lineNum < 0:
            lineNum = 0
            break
        line = content[lineNum]
        for c in line[::-1]:
            if c=='{':
                count -= 1
                if count==0:
                    break
            if c=='}':
                count += 1
    sStart = lineNum+1
    return (sStart, sEnd)
```

This part is for finding the smallest enclosing scope.

The main idea is to find the nearest “{” and “}” for current line. But there is one thing need be careful, the nearest “{” or “}” is not the enclosing scope. For example:

```

47 53      */
48 54      public static String normalize(String path, boolean replaceBackSlash) {
49 55
50 -      if (path == null)
56 +      if (path == null) {
51 57          return null;
58 +      }
52 59
53 60          // Create a place for the normalized path
54 61          String normalized = path;
55 62
56 63          if (replaceBackSlash && normalized.indexOf('\\') >= 0)
57 64              normalized = normalized.replace('\\', '/');
58 65
59 -      if (normalized.equals("/."))
60 -      return "/";
61 -

```

If we want to find line 59 scope, we suppose to find line 48 not line 50-51. In order to do so, I use parameter “count” to find the 1<sup>st</sup> unpaired “{” and “}”. For example, when I find upper scope, I need find the 1<sup>st</sup> unpaired “{”. I set default count equal 1. When I find a “}”, the count will increase one, when I find a “{”, the count will decrease one, when count equal 0, it means I find the 1<sup>st</sup> unpaired “{” for upper scope.

## Identify the latest commit

```

In [42]: def latestCommit(blames):
        commits = []
        times = []
        for b in blames:
            tmp = b.split()
            commits += [tmp[0]]
            times += [int(tmp[3])]
        return commits[np.argmax(times)]

```

Then we need find out the more recent commit, which is used for finding VCC.

### Find VCC

```

def findVCC(repo_path, fixing_commit):
    repo = Repo(repo_path)
    HEAD = fixing_commit
    PREV = fixing_commit + '^'
    files = repo.git.diff('--name-only', PREV, HEAD).splitlines()
    commitsFound = []

    for file in files:
        try:
            content = repo.git.show(PREV+' '+file).splitlines()
        except:
            continue
        dlines = repo.git.diff('-U0', PREV, HEAD, file).splitlines()
        sumlines = sumlines(dlines)
        for line in sumlines:
            (dStart, dLength, aStart, aLength) = pSummary(line)
            if dLength > 0:
                blames = repo.git.blame('-w', '--date=unix', '-e', '-f', '-L '+str(dStart) + ', ' + str(dStart + dLength - 1), PREV,
                commit = latestCommit(blames)
                commitsFound += [commit]*dLength
            if aLength > 0:
                (sStart, sEnd) = findScope(dStart, dLength, aStart, aLength, content)
                blames = repo.git.blame('-w', '--date=unix', '-e', '-f', '-L '+str(sStart) + ', ' + str(sEnd), PREV, file).splitlines()
                commit = latestCommit(blames)
                commitsFound += [commit]*aLength
    VCC = max(set(commitsFound), key=commitsFound.count)
    return VCC

```

In this part, when there is a delete line or an add line, I use git blame with scope parameter to find out the latest commit for each line changed and save all result in a set. Then find the commit that appeared the most times, that is the VCC we want to find.

- **VCC Result**

**Case 1 CVE-2015-5174, VCC is d0e4c578f6.**

**Case 1 CVE-2015-5174**

```
44]: remote_link = "https://github.com/apache/tomcat80"
local_link = "~/Documents/tomcat80"
if not os.path.isdir(local_link):
    Repo.clone_from(remote_link, local_link, progress = Progress())
fixing_commit = "2fc9d03ffbc3fe7eabfd272380807ac0ddcf748d"
vcc = find_vcc(local_link, fixing_commit)
print("\nCase 1 repository: %s, commit: %s" % (remote_link, fixing_commit))
print("\nVCC is ", vcc)
```

Case 1 repository: <https://github.com/apache/tomcat80>, commit: 2fc9d03ffbc3fe7eabfd272380807ac0ddcf748d

VCC is d0e4c578f6

**Case 2 CVE-2018-1325, VCC is a7b9f7961e.**

**Case 2 CVE-2018-1325**

```
45]: remote_link = "https://github.com/sebfz1/wicket-jquery-ui"
local_link = "~/Documents/wicket-jquery-ui"
if not os.path.isdir(local_link):
    Repo.clone_from(remote_link, local_link, progress = Progress())
fixing_commit = "2fec03dbe2f6e8808f4bdc6b3195dff3e44f520"
vcc = find_vcc(local_link, fixing_commit)
print("\nCase 2 repository: %s, commit: %s" % (remote_link, fixing_commit))
print("\nVCC is ", vcc)
```

Case 2 repository: <https://github.com/sebfz1/wicket-jquery-ui>, commit: 2fec03dbe2f6e8808f4bdc6b3195dff3e44f520

VCC is a7b9f7961e

**Case 3 CVE-2013-7251, VCC is 96ca7e5ac0.**

**Case 3 CVE-2013-7251**

```
46]: remote_link = "https://github.com/micromata/projectforge-webapp"
local_link = "~/Documents/projectforge-webapp"
if not os.path.isdir(local_link):
    Repo.clone_from(remote_link, local_link, progress = Progress())
fixing_commit = "422de35e3c3141e418a73bfb39b430d5fd74077e"
vcc = find_vcc(local_link, fixing_commit)
print("\nCase 3 repository: %s, commit: %s" % (remote_link, fixing_commit))
print("\nVCC is ", vcc)
```

Case 3 repository: <https://github.com/micromata/projectforge-webapp>, commit: 422de35e3c3141e418a73bfb39b430d5fd74077e

VCC is 96ca7e5ac0

- **Analysis for VCCs: I used exercise 2 script to output the results. No explanation here. All result can be checked in Exercise3\_Analysis\_Tianfang\_Wang\_a1788845.ipynb.**

**CASE 1 CVE-2015-5174**

## Case 1 CVE-2015-5174

```
In [82]: repo1 = Repo("~/Documents/tomcat80")
        fixing_commit_1 = "d0e4c578f6"
```

**Q(a) What was the message and title of the fixing commit? Was there any mention of fixing a bug or vulnerability?**

```
In [83]: res1 = repo1.git.log(-1, '--pretty=%B', fixing_commit_1)
        print("Title:", res1)

Title: Refactor to remove the circular dependency between o.a.catalina and
o.a.naming

git-svn-id: https://svn.apache.org/repos/asf/tomcat/trunk@1239048 13f79535-47bb-0310-9956-ffa450edef68
```

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

```
In [84]: res2 = repo1.git.diff('--name-only', fixing_commit_1, fixing_commit_1 + '^').splitlines()
        print("Number of affected files: %d"%len(res2))

Number of affected files: 10
```

**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.**

```
In [85]: res3 = repo1.git.diff('--dirstat', fixing_commit_1, fixing_commit_1 + '^').splitlines()
        print("Number of affected directories: %d"%len(res3))

Number of affected directories: 6
```

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

```
In [86]: res4 = repo1.git.show(fixing_commit_1).splitlines()
        res4 = [line for line in res4 if re.compile('^-|^\s+$').match(line)]
        print("Number of deleted lines (including comments and blank lines): %d"%len(res4))

Number of deleted lines (including comments and blank lines): 71
```

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

```
In [87]: res5 = repo1.git.show(fixing_commit_1).splitlines()
        res5 = [line for line in res5 if re.compile('^[^+]\s+$').match(line)]
        print("Number of added lines (including comments and blank lines): %d"%len(res5))

Number of added lines (including comments and blank lines): 157
```

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

```
In [88]: res6 = repo1.git.show('--ignore-blank-lines', fixing_commit_1).splitlines()
        res6 = [line for line in res6 if re.compile('^-|^\s+$').match(line)]
        res6 = [line for line in res6 if not re.compile('^[^+]\s+$').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): 49
```

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

```
In [89]: res7 = repo1.git.show('--ignore-blank-lines', fixing_commit_1).splitlines()
        res7 = [line for line in res7 if re.compile('^[^+]\s+$').match(line)]
        res7 = [line for line in res7 if not re.compile('^[^+]\s+$').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): 69
```

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

```
In [90]: 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/catalina/core/ApplicationContext.java , the days between the current fixing commit and the previous commit is 79.5 days  
For java/org/apache/catalina/servlets/WebdavServlet.java , the days between the current fixing commit and the previous commit is 12.8 days  
For java/org/apache/catalina/ssi/SSIServletExternalResolver.java , the days between the current fixing commit and the previous commit is 101.6 days  
For java/org/apache/catalina/ssi/SSIServletRequestUtil.java , the days between the current fixing commit and the previous commit is 101.6 days  
For java/org/apache/catalina/util/RequestUtil.java , the days between the current fixing commit and the previous commit is 12.8 days  
For java/org/apache/naming/resources/FileDialogContext.java , the days between the current fixing commit and the previous commit is 33.6 days  
java/org/apache/tomcat/util/http/RequestUtil.java: N/A This file doesn't have previous commit  
For res/checkstyle/org-import-control.xml , the days between the current fixing commit and the previous commit is 45.5 days  
For test/org/apache/catalina/util/TestRequestUtil.java , the days between the current fixing commit and the previous commit is 101.6 days  
test/org/apache/tomcat/util/http/TestRequestUtil.java: N/A This file doesn't have previous commit

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

```
In [91]: 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/catalina/core/ApplicationContext.java has been modified 71 times  
java/org/apache/catalina/servlets/WebdavServlet.java has been modified 44 times  
java/org/apache/catalina/ssi/SSIServletExternalResolver.java has been modified 20 times  
java/org/apache/catalina/ssi/SSIServletRequestUtil.java has been modified 9 times  
java/org/apache/catalina/util/RequestUtil.java has been modified 25 times  
java/org/apache/naming/resources/FileDialogContext.java has been modified 40 times  
java/org/apache/tomcat/util/http/RequestUtil.java has been modified 1 times  
res/checkstyle/org-import-control.xml has been modified 8 times  
test/org/apache/catalina/util/TestRequestUtil.java has been modified 9 times  
test/org/apache/tomcat/util/http/TestRequestUtil.java has been modified 9 times

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

```
n [92]: 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/catalina/core/ApplicationContext.java has been modified by following developers:  
Remy Maucherat  
Mark Thomas  
Konstantin Kolinko  
Filip Hanik  
Mladen Turk  
Rainer Jung  
java/org/apache/catalina/servlets/WebdavServlet.java has been modified by following developers:  
Tim Whittington  
Remy Maucherat  
Mark Thomas  
Konstantin Kolinko  
Mladen Turk  
Rainer Jung  
java/org/apache/catalina/ssi/SSIServletExternalResolver.java has been modified by following developers:  
Konstantin Kolinko  
Mark Thomas  
Remy Maucherat  
Mladen Turk  
java/org/apache/catalina/ssi/SSIServletRequestUtil.java has been modified by following developers:  
Mark Thomas  
Remy Maucherat  
Mladen Turk  
java/org/apache/catalina/util/RequestUtil.java has been modified by following developers:  
Remy Maucherat  
Mark Thomas  
Konstantin Kolinko  
Filip Hanik  
Mladen Turk  
Rainer Jung  
java/org/apache/naming/resources/FileDialogContext.java has been modified by following developers:  
Remy Maucherat  
Timothy A. Funk  
Mark Thomas  
Konstantin Kolinko

**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?**

```
[93]: 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))))

Tim Whittington made 22 commits
Remy Maucherat made 434 commits
Timothy A. Funk made 61 commits
Mark Thomas made 4729 commits
Konstantin Kolinko made 675 commits
Filip Hanik made 1000 commits
Mladen Turk made 124 commits
Rainer Jung made 280 commits
```

## CASE 2 CVE-2018-1325

### Case 2 CVE-2018-1325

```
In [94]: repo2 = Repo("../Documents/wicket-jquery-ui")
fixing_commit_2 = "a7b9f7961e"
```

**Q(a) What was the message and title of the fixing commit? Was there any mention of fixing a bug or vulnerability?**

```
In [95]: res1 = repo2.git.log(-1, '--pretty=%B', fixing_commit_2)

print("Title:", res1)

Title: Added wysiwygEditor
```

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

```
In [96]: res2 = repo2.git.diff('--name-only', fixing_commit_2, fixing_commit_2 + '^').splitlines()
print("Number of affected files: %d"%len(res2))

Number of affected files: 25
```

**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.**

```
In [97]: res3 = repo2.git.diff('--dirstat', fixing_commit_2, fixing_commit_2 + '^').splitlines()
print("Number of affected directories: %d"%len(res3))

Number of affected directories: 4
```

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

```
In [98]: res4 = repo2.git.show(fixing_commit_2).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): 2
```



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

```
In [99]: res5 = repo2.git.show(fixing_commit_2).splitlines()
res5 = [line for line in res5 if re.compile('^\s+$|^\s+$').match(line)]
print("Number of added lines (including comments and blank lines): %d"%len(res5))
```

Number of added lines (including comments and blank lines): 4478

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

```
In [100]: res6 = repo2.git.show('--ignore-blank-lines', fixing_commit_2).splitlines()
res6 = [line for line in res6 if re.compile('^\s+$|^\s+$').match(line)]
res6 = [line for line in res6 if not re.compile('^\s+$|^\s+$').match(line)]
res6 = [line for line in res6 if not re.compile('^\s+$|^\s+$').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res6))
```

Number of added lines (excluding comments and blank lines): 0

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

```
In [101]: res7 = repo2.git.show('--ignore-blank-lines', fixing_commit_2).splitlines()
res7 = [line for line in res7 if re.compile('^\s+$|^\s+$').match(line)]
res7 = [line for line in res7 if not re.compile('^\s+$|^\s+$').match(line)]
res7 = [line for line in res7 if not re.compile('^\s+$|^\s+$').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res7))
```

Number of added lines (excluding comments and blank lines): 3515

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

```
In [102]: for file in repo2.git.diff('--name-only', fixing_commit_2, fixing_commit_2 + '^').splitlines():
# print(file)
res8 = repo2.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)
```

```
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygBehavior.java: N/A This file
doesn't have previous commit
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.html: N/A This file do
esn't have previous commit
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.java: N/A This file do
esn't have previous commit
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.html:
N/A This file doesn't have previous commit
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.java:
N/A This file doesn't have previous commit
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/IWysiwygToolbar.java: N/A Th
is file doesn't have previous commit
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-combined.no-icons.
min.css: N/A This file doesn't have previous commit
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-responsive.min.cs
s: N/A This file doesn't have previous commit
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/editor.css: N/A This file d
oesn't have previous commit
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/font-awesome.css: N/A This
```

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

```
In [103]: for file in repo2.git.diff('--name-only', fixing_commit_2, fixing_commit_2 + '^').splitlines():
res9 = repo2.git.log('--follow', '--pretty=oneline', file).splitlines()
print(" %s has been modified %d times"%(file, len(res9)))

wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygBehavior.java has been modified
1 times
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.html has been modified 4
times
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.java has been modified 1
times
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.html has
been modified 1 times
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.java has
been modified 5 times
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/IWysiwygToolbar.java has been
modified 7 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-combined.no-icons.m
in.css has been modified 1 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-responsive.min.css
has been modified 1 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/editor.css has been modified
5 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/font-awesome.css has been mod
ified 1 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/font/fontawesome-webfont.eot has
been modified 1 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/font/fontawesome-webfont.ttf has
been modified 1 times
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/font/fontawesome-webfont.woff has
been modified 1 times
```



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

```
In [104]: for file in repo2.git.diff('--name-only',fixing_commit_2, fixing_commit_2 + '^').splitlines():
          print(" %s has been modified by following developers:%file)
          res10 = repo2.git.log('--follow', '--pretty=%aN', file).splitlines()
          for dev in set(res10):
              print(" %s"%dev)

wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygBehavior.java has been modified
by following developers:
Sebastien
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.html has been modified b
y following developers:
Sebastien
Sebastien Briquet
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/WysiwygEditor.java has been modified b
y following developers:
Sebastien
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.html has
been modified by following developers:
Sebastien
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/DefaultWysiwygToolbar.java has
been modified by following developers:
Sebastien
Sebastien Briquet
wicket-jquery-ui-plugins/src/main/java/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/toolbar/IWysiwygToolbar.java has been
modified by following developers:
Sebastien
Sebastien Briquet
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-combined.no-icons.m
in.css has been modified by following developers:
Sebastien
wicket-jquery-ui-plugins/src/main/resources/com/googlecode/wicket/jquery/ui/plugins/wysiwyg/css/bootstrap-combined.no-icons.m
in.css
```

### 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?

```
In [105]: devs = []
          for file in repo2.git.diff('--name-only',fixing_commit_2, fixing_commit_2 + '^').splitlines():
              devs += repo2.git.log('--follow', '--pretty=%aN', file).splitlines()
          res11 = list(set(devs))
          log = repo2.git.log('--pretty=%aN')
          for dev in res11:
              print(" %s made %d commits"%(dev, len(re.findall(dev,log))))

Sebastien made 186 commits
Sebastien Briquet made 175 commits
```

## CASE 3 CVE-2013-7251

### Case 3 CVE-2013-7251

```
In [106]: repo3 = Repo("~/Documents/projectforge-webapp")
          fixing_commit_3 = "96ca7e5ac0"
```

### Q(a) What was the message and title of the fixing commit? Was there any mention of fixing a bug or vulnerability?

```
In [107]: res1 = repo3.git.log(-1, '--pretty=%B', fixing_commit_3)
          print("Title:",res1)

Title: PF-396: Disable autocompletion in all forms by default.

git-svn-id: https://svn.micromata.de/svn/projectforge/webapp/trunk@14054 be79319d-3443-0410-9b0a-b0c1b946f33b
```

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

```
In [108]: res3 = repo3.git.diff('--name-only', fixing_commit_3, fixing_commit_3 + '^').splitlines()
          print("Number of affected files: %d"%len(res3))

Number of affected files: 4
```

### 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.

```
In [109]: res3 = repo3.git.diff('--dirstat', fixing_commit_3, fixing_commit_3 + '^').splitlines()
          print("Number of affected directories: %d"%len(res3))

Number of affected directories: 3
```

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

```
In [110]: res4 = repo3.git.show(fixing_commit_3).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): 4

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

```
In [111]: res5 = repo3.git.show(fixing_commit_3).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): 4

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

```
In [112]: res6 = repo3.git.show('--ignore-blank-lines', fixing_commit_3).splitlines()
res6 = [line for line in res6 if re.compile('^[^~]').match(line)]
res6 = [line for line in res6 if not re.compile('^[^~]$|^[^~]').match(line)]
res6 = [line for line in res6 if not re.compile('^[^~]$|^[^~]').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res6))
```

Number of added lines (excluding comments and blank lines): 4

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

```
In [113]: res7 = repo3.git.show('--ignore-blank-lines', fixing_commit_3).splitlines()
res7 = [line for line in res7 if re.compile('^[^~]').match(line)]
res7 = [line for line in res7 if not re.compile('^[^~]$|^[^~]').match(line)]
res7 = [line for line in res7 if not re.compile('^[^~]$|^[^~]').match(line)]
print("Number of added lines (excluding comments and blank lines): %d"%len(res7))
```

Number of added lines (excluding comments and blank lines): 4

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

```
In [114]: for file in repo3.git.diff('--name-only', fixing_commit_3, fixing_commit_3 + '^').splitlines():
res8 = repo3.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 src/main/java/org/projectforge/web/dialog/ModalDialog.html , the days between the current fixing commit and the previous c
ommit is 2.0 days
For src/main/java/org/projectforge/web/fibu/RechnungCostEditTablePanel.html , the days between the current fixing commit and t
he previous commit is 36.0 days
For src/main/java/org/projectforge/web/mobile/AbstractMobileEditPage.html , the days between the current fixing commit and the
previous commit is 36.0 days
For src/main/java/org/projectforge/web/mobile/AbstractMobileListPage.html , the days between the current fixing commit and the
previous commit is 36.0 days
>
```

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

```
In [115]: for file in repo3.git.diff('--name-only', fixing_commit_3, fixing_commit_3 + '^').splitlines():
res9 = repo3.git.log('--follow', '--pretty=oneline', file).splitlines()
print(" %s has been modified %d times"%(file, len(res9)))

src/main/java/org/projectforge/web/dialog/ModalDialog.html has been modified 8 times
src/main/java/org/projectforge/web/fibu/RechnungCostEditTablePanel.html has been modified 3 times
src/main/java/org/projectforge/web/mobile/AbstractMobileEditPage.html has been modified 11 times
src/main/java/org/projectforge/web/mobile/AbstractMobileListPage.html has been modified 12 times
```

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

```
In [116]: for file in repo3.git.diff('--name-only',fixing_commit_3, fixing_commit_3 + '^').splitlines():
          print(" %s has been modified by following developers:"%file)
          res10 = repo3.git.log('--follow', '--pretty=%aN', file).splitlines()
          for dev in set(res10):
              print("    %s"%dev)

src/main/java/org/projectforge/web/dialog/ModalDialog.html has been modified by following developers:
kai
src/main/java/org/projectforge/web/fibu/RechnungCostEditTablePanel.html has been modified by following developers:
kai
src/main/java/org/projectforge/web/mobile/AbstractMobileEditPage.html has been modified by following developers:
kai
src/main/java/org/projectforge/web/mobile/AbstractMobileListPage.html has been modified by following developers:
kai
```

### 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?

```
In [117]: devs = []
          for file in repo3.git.diff('--name-only',fixing_commit_3, fixing_commit_3 + '^').splitlines():
              devs += repo3.git.log('--follow', '--pretty=%aN', file).splitlines()
          res11 = list(set(devs))
          log = repo3.git.log('--pretty=%aN')
          for dev in res11:
              print(" %s made %d commits"%(dev, len(re.findall(dev,log))))

kai made 4013 commits
```

---

- **Git blame with different parameters.**

Parameters explanation:

- w is ignore whitespace when comparing two versions.
- M is detect moved or copied lines within a file.
- C is detect lines moved or copied from other files that were modified in the same commit.

For my 3 VCCs results, I got same result by using those different combinations of parameters.

- **Similarities and differences**

For similarities: Most commits have same author and same files.

For differences: For each files, they changed different lines, modify times are different. Some commits fixed different bugs.