

**File Name: Exercise 2 Report**

**Student Name: Wuli Zuo**

**Student ID: a1785343**

**Vulnerabilities Assigned: CVE-2018-11087, CVE-2018-11797, CVE-2017-5647**

**GitHub account: DoriaSummer**

**Link to the repository of Exercise 2: [https://github.com/DoriaSummer/sse\\_exercise2](https://github.com/DoriaSummer/sse_exercise2)**

In Exercise 2, bash commands and Python are used to analyse the fix commits. The structure of the Python script is described in Section 1 of this report, and each Git command and corresponding Python codes are explained in Section 2. Since only one of the commits is taken as an example in Section 2, the output of other commits is included in Section 3.

## 1. Structure of the automated script

The automated script is written with Python. Functions from `GitPython` library are used to analyse each fix commit.

First, `Repo.clone_from()` is used to clone the repository to local directory, and then the function `analyse()` is called to analyse each fix commit and output the answers to the questions in Task 3.

In the function `analyse()`, first reset to the given commit, and then functions, including `git.log()` and `git.show()`, are used to answer each question of Task 3.

**Specific comments are included in the source code file.** It is most recommended to read the code to get details: [https://github.com/DoriaSummer/sse\\_exercise2/blob/master/Analyse.py](https://github.com/DoriaSummer/sse_exercise2/blob/master/Analyse.py).

## 2. Explanations of the Git commands and Python codes

This part aims to explain each Git command used in Exercise 2, and corresponding Python codes are showed as well.

To illustrate the whole exercise process more specifically, the given commit of Case 1: CVE-2018-11087 is taken as an example here:

- CVE-ID: CVE-2018-11087
- Link: <https://github.com/spring-projects/spring-amqp>
- Fixing Commit: 444b74e95bb299af5e23ebf006fbb45d574fb95 (the given commit)

### 2.1 Preparation

To start, clone the repository and reset to the target commit.

(1) Clone the repository `spring-amqp` to local directory: `~/Workspace/spring-amqp`

Git command:

---

```
git clone https://github.com/spring-projects/spring-amqp ~/Workspace/spring-amqp
```

---

Python code:

---

```
remote_link = "https://github.com/spring-projects/spring-amqp"
local_link = "../spring-amqp"
if not os.path.isdir(local_link):
    Repo.clone_from(remote_link, local_link, progress=Progress())
```

---

(2) (If use bash command in Terminal) Go to local directory of the repository:  
`~/Workspace/spring-amqp`

Git command:

---

```
cd Workspace/spring-amqp
```

---

(3) Reset to the given commit: `444b74e95bb299af5e23ebf006fbb45d574fb95` with option `hard`

Git command:

---

```
git reset --hard 444b74e95bb299af5e23ebf006fbb45d574fb95
```

---

Python code:

---

```
# Reset to given commit
output_0 = repo.git.reset('--hard', fixing_commit)
print("Current fix commit:\n%s" % output_0)
```

---

## 2.2 Task 3

The commands used to seek the answer to each question are listed and explained as follows.

- a. What was the message and title of the fixing commit? Was there any mention of fixing a bug or vulnerability?

The command `git log` with option `-1` is used here, which means to get only one log, starting from current commit. The option `format=%B` is used here to limit the output format as raw body.

From the output, it was mentioned in the message that the commit was the fix of a bug numbered 830.

Git command:

---

```
git log -1 --format=%B
```

---

Python code:

---

```
# Question a: get commit message and title

output_a = repo.git.log(-1, '--format=%B')

print("\n(a). Commit message and title:\n%s" % output_a)
```

---

Output:

---

```
(a). Commit message and title:
```

---

```
AMQP-830 Enable Hostname Verification by default
```

JIRA: <https://jira.spring.io/browse/AMQP-830>

---

- b. How many total files were affected in the fixing commit?

The command `git show` is used to find the changes of the given commit. `HEAD` can be used as parameter refers to current commit, but since the given commit was reset as default, `HEAD` is omitted here. The option `--name-only --format=""` is used to limit the output to show name of the affected files, and `wc -l` is used to count lines, so that the output is the number of lines of the command left to the piping symbol `|`, which is the number of files.

The output shows that 3 files were affected.

Git command:

---

```
git show --name-only --format="" | wc -l
```

---

Python code:

---

```
# Question b: get number of affected files

output_b = repo.git.show('--name-only', '--format=').splitlines()

print("(b). Number of affected files: %d" % len(output_b))
```

---

Output:

---

(b). Number of affected files: 3

---

- 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.

Similar to b, the difference is that another option `--dirstat` is used to count at level of directory, and the output is the number of lines, that is the number of directories affected.

The output shows that 2 directories were affected in total.

Git command:

---

```
git show --dirstat --format="" | wc -l
```

---

Python code:

---

```
# Question c: get number of affected directories

output_c = repo.git.show('--dirstat', '--format=').splitlines()

print("(c). Number of affected directories: %d" % len(output_c))
```

---

Output:

---

```
(c). Number of affected directories: 2
```

---

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

Similar to b and c, `git show` command is used. In order to calculate the lines deleted, the lines starting with `'-'` need to be counted, comments and blank lines need to be included, but the lines showing the affected file directory and name, starting with `'--'` should be excluded:

Excluded lines:

---

```
--- b/build.gradle

--- b/spring-rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnect
ionFactoryBean.java

--- b/src/reference/asciidoc/whats-new.adoc
```

---

Grep command is used here to search the lines matching the pattern defined by the regular expression `'^[^-\]|^-\s*$'`, in which the first part represents a line starting with a `'-'` but not followed by a `'-'` and the right part represents a line starting with a `'-'` and

followed by arbitrary number of spaces. Comments and blank lines deleted is included in this scenario. The result is the number of lines, showing that 12 lines were deleted in total.

Git command:

---

```
git show | grep '^-[^-]\\|^-\\s*$' | wc -l
```

---

Python code:

---

```
# Question d: get number of deleted lines (including comments and blank lines)

lines = repo.git.show().splitlines()

# Lines start with '-', but not followed by '-', can be followed arbitrary number of
blanks

output_d = [line for line in lines if re.compile('^-[^-]|^-\\s*$').match(line)]

print("(d). Number of deleted lines (including comments and blank lines): %d" %
len(output_d))
```

---

Output:

---

```
(d). Number of deleted lines (including comments and blank lines): 12
```

---

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

Similar to d, just change the pattern into `'^+[^+]|^+\\s*$'`, representing a line starting with a '+' but not followed by a '+' or followed by arbitrary number of spaces, so that the lines of the affected file directory and name is excluded, and space lines is included.

The output shows that there were 40 lines added.

Git command:

---

```
git show | grep '^+[^+]|^+\\s*$' | wc -l
```

---

Python code:

---

```
# Question e: get number of added lines (including comments and blank lines)

lines = repo.git.show().splitlines()

# Lines start with '+', but not followed by '+', can be followed arbitrary number of bla
nks

output_e = [line for line in lines if re.compile('^+[^+]|^+\\s*$').match(line)]
```

---

---

```
print("(e). Number of added lines (including comments and blank lines): %d" % len(output_e))
```

---

Output:

---

```
(e). Number of added lines (including comments and blank lines): 40
```

---

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

Based on d, what need to be done here is to exclude comments and blank lines from the pattern, including block comments, line comments and blank lines:

Comments and blank lines patterns need to be excluded:

---

```
- /**
-  * This is a block comment, including two line patterns.
-  */
- // This is a line comment. The next line is a blank line, might starts with a hyphen,
followed by arbitrary number of spaces
-
```

---

This is accomplished by excluding the above patterns using the command `grep -v` step by step. The first `grep -v` command excludes the lines starting with a '-', followed by arbitrary number of spaces and then a '/', that is a line comment or the starting line of a block comment; the second `grep -v` excludes the lines starting with a '-', followed by arbitrary number of spaces and then an '\*', which are other lines of block comments; and the last `grep -v` excludes all the blank lines.

The output shows that 12 lines of code (excluding comments and blank lines) were deleted.

Git command:

---

```
git show | grep '^-[^-]\|^-\s*$' | grep -v '^-\s*/' | grep -v '^-\s*\*' | grep -v '^-\s*$'
| wc -l
```

---

Python code:

---

```
# Question f: get number of deleted lines (excluding comments and blank lines)

lines = repo.git.show().splitlines()
```

---

---

```

# Lines start with '-', but not followed by '-'; excluding /*, *, */ and blank
Lines

lines = [line for line in lines if re.compile('^-[^-]|\s*$').match(line)]
lines = [line for line in lines if not re.compile('^\s*\/').match(line)]
lines = [line for line in lines if not re.compile('^\s*\*').match(line)]
output_f = [line for line in lines if not re.compile('^\s*$').match(line)]

print("(f). Number of deleted lines (including comments and blank lines): %d" %
len(output_f))

```

---

Output:

---

```

(f). Number of deleted lines (including comments and blank lines): 12

```

---

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

Similar to f, comments and blank lines need to be excluded from the result of by using `grep -v`. After changing '-' into '+', all the patterns are same with those in g.

The output shows that 24 lines of code (excluding comments and blank lines) were added.

Git command:

---

```

git show | grep '^+[^+]\|\s*$' | grep -v '^\s*\/' | grep -v '^\s*\*' | grep -v '^\s*$'
| wc -l

```

---

Python code:

---

```

# Question g: get number of added Lines (excluding comments and blank Lines)

lines = repo.git.show().splitlines()

# Lines start with '+', but not followed by '+'; excluding /*, *, */ and blank
Lines

lines = [line for line in lines if re.compile('^+[^+]\|\s*$').match(line)]
lines = [line for line in lines if not re.compile('^\s*\/').match(line)]
lines = [line for line in lines if not re.compile('^\s*\*').match(line)]
output_g = [line for line in lines if not re.compile('^\s*$').match(line)]

print("(g). Number of added lines (including comments and blank lines): %d" %
len(output_g))

```

---

Output:

---

```
(g). Number of added lines (including comments and blank lines): 24
```

---

- h. How many days were between the current fixing commit and the previous commit of each affected file?

For each file affected, print the file name on screen, then use `git log -2` to get current commit and the previous commit log respectively, `pretty=%ct` is used to limit the output format as commit time, compute the difference between two commit times and convert the result unit from in seconds to in days.

The output is the number of days between two commits of each affected file, 8.01 days, 284.94 days and 13.13 days respectively. The average number of days between two commits of each file is 102.03.

Git command:

---

```
for file in `git show --name-only --format=""`; do echo $file; git log -2 --format=%ct $file | awk 'NR==1{ct=$1;next} {print (ct-$1)/(60*60*24);} END{ }'; done
```

---

Python code:

---

```
# Question h: number of days between two commit

print("(h). Number of days between the current fixing commit and the previous commit: ")

total_days = 0

for file in repo.git.show('--name-only', '--format=').splitlines():

    # Check if a file is a new added file in current commit

    output_h = repo.git.log(-2, '--format=%ct', file).splitlines()    if(len(output_h) =

= 1):

        print("    %s: new added file 0")

    else:

        # convert unit from seconds to days

        days = (float(output_h[0])-float(output_h[1]))/(60*60*24)

        total_days += days

        print("    %s: %.2f" % (file, days))

avg_days = total_days / len(output_b)

print("    The average number of days between two commits: %.2f" % avg_days)
```

---



Output:

---

(h). Number of days between the current fixing commit and the previous commit:

```
build.gradle: 8.01
spring-rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnectionFactoryBean.java: 284.94
src/reference/asciidoc/whats-new.adoc: 13.13
The average number of days between two commits: 102.03
```

---

- i. How many time has each affected file of the current fixing commit been modified in the past since their creation (including rename of the file)?

For each file affected, print the file name on screen, use the command `git log` with option `--follow` to list the history of the file, and `--format=oneline` is used here to output the history in format of one line, calculate the commit times by counting the lines of the log.

The output shows that three affected files have been modified 146 times, 17 times and 108 times respectively. The average times of each file has been modified is 90.33.

Git command:

---

```
for file in `git show --name-only --format=""; do echo $file; git log --follow --format=oneline $file | wc -l; done
```

---

Python code:

---

```
# Question i: number of times each affected file has been modified
print("(i). Number of times each file has been modified: ")
total_times = 0
for file in repo.git.show('--name-only', '--format=').splitlines():
    output_i = repo.git.log('--follow', '--format=oneline', file).splitlines()
    total_times += len(output_i)
    print("    %s: %d" % (file, len(output_i)))
avg_times = total_times / len(output_b)
print("    The average number of modified times: %.2f" % avg_times)
```

---

Output:

---

(i). Number of times each file has been modified:

---

---

```
build.gradle: 146
spring-rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnecti
onFactoryBean.java: 17
src/reference/asciidoc/whats-new.adoc: 108
The average number of modified times: 90.33
```

---

j. Which developers have modified each affected file since its creation?

For each file affected, print the file name on screen, use the command `git log` with option `--format="%aN"` to list all the authors (respecting mailmap), `sort -u` is used here to keep unique of each author name and to sort the authors in ascending order.

Git command:

---

```
for file in `git show --name-only --format=""`; do echo $file; git log --format=" %aN"
$file |sort -u; done
```

---

Python code:

---

```
# Question j: developers who have modified each file

print("(j). Developers who have modified each file: ")

total_authors = 0

for file in repo.git.show('--name-only', '--format=').splitlines():
    print("    Developers who have modified file: %s " % file)
    authors = repo.git.log('--follow', '--format=%aN', file).splitlines()
    for author in set(authors):
        print("        %s"%author)
        total_authors = total_authors + 1
    avg_authors = total_authors / len(output_b)

print("    The average number of developers modified each file: %.2f" % avg_authors)
```

---

Output:

---

```
(j). Developers who have modified each file:

Developers who have modified file: build.gradle

    Andy Wilkinson

    Mohammad Hewedy
```

---

Arnaud Cogoluègnes
Aleksander Morozov
Stéphane Nicoll
Jon Brisbin
Andreas Asplund
Gary russell
Gunnar Hillert
Alex Panchenko
Gary Russell
Artem Bilan
Developers
who
have
modified
file:
spring- rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnectionFactoryBea n.java

Dominique Villard

Arnaud Cogoluègnes

Gary Russell

Heath Abelson

Zachary DeLuca

Mohammad Hewedy

Arnaud Cogoluègnes

Alex Panchenko

Gary Russell

Steve Powell

Ernest Sadykov

Artem Bilan

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?

Get the list of the author's name in similar way as illustrated in j, but do `sort -u` overall at last instead of respectively, because there could be some developers made fix commits for more than one file. Then for each author in the list, print the author's name on screen, match the author's name in logs and calculate the number of lines, that is the commits of the current author submitted.

The output shows that the total number of developers who have modified the files is 19, among which some are quite experienced with 623 commits at most, while others may be new ones with only 1 commit.

Git command:

---

```
for file in `git show --name-only --format=""; do git log --format=%aN $file; done | sort -u | while read auth; do echo -n $auth ": "; git log --format=%aN | grep "$auth" | wc -l; done
```

---

Python code:

---

```
# Question k: number of commits each developer has made

print("(k). Number of commits each developer has made: ")

authors = set()

for file in repo.git.show('--name-only', '--format=').splitlines():
    authors.update(repo.git.log('--format=%aN', file).splitlines())

authors = list(authors)

authors.sort()

for author in authors:

    commits_time = len(re.findall(author,repo.git.log('--format=%aN')))

    if commits_time >= 100:

        print("    %s: %d This is an experienced developer." % (author, commits_time))

    else:

        print("    %s: %d This is an new developer." % (author, commits_time))

print("    Total number of developers modified the affected files: %d" % len(authors))
```

---

Output:

---

```
(k). Number of commits each developer has made:

Aleksander Morozov: 3 This is an new developer.

Alex Panchenko: 17 This is an new developer.
```

---

---

Andreas Asplund: 1 This is an new developer.

Andy Wilkinson: 5 This is an new developer.

Arnaud Cogoluègues: 5 This is an new developer.

Artem Bilan: 148 This is an experienced developer.

Dominique Villard: 1 This is an new developer.

Ernest Sadykov: 1 This is an new developer.

Gary Russell: 623 This is an experienced developer.

Gary russell: 1 This is an new developer.

Gunnar Hillert: 36 This is an new developer.

Heath Abelson: 1 This is an new developer.

Jon Brisbin: 1 This is an new developer.

Mohammad Hewedy: 1 This is an new developer.

Steve Powell: 1 This is an new developer.

Stéphane Nicoll: 1 This is an new developer.

Zachary DeLuca: 1 This is an new developer.

dreis2211: 3 This is an new developer.

harrydevnull: 1 This is an new developer.

Total number of developers modified the affected files: 19

---

### 3. Output of other commits

#### 3.1 Case 1:

- CVE-ID: CVE-2018-11087
- Link: <https://github.com/spring-projects/spring-amqp>
- Fixing Commit: f8e7732ce69e5f3e591700bebbf00682ce7ab231 (the identified commit)
- Notes: As mentioned in Exercise 1, a complete fix to the vulnerability is provided by combining the provided commit and identified commit together, so both commits are analysed in Exercise 2.

Output of case 1-identified commit:

---

Analyses of repo: <https://github.com/spring-projects/spring-amqp>, commit: f8e7732ce69e5f3e591700bebbf00682ce7ab231

Current fix commit:

HEAD is now at f8e7732c AMQP-689: Support skipServerCertificateValidation

---

---

(a). Commit message and title:

AMQP-689: Support skipServerCertificateValidation

JIRA: <https://jira.spring.io/browse/AMQP-689>

Fixes GH-522 (<https://github.com/spring-projects/spring-amqp/issues/522>)

Instead of using the unsafe

useSslProtocol() and passing on the rabbit-amqp layer

we delegate the trust mechanism to Java's default way.

removed the redundant try catch blocks

test cases added a matcher for sslcontext

updated author

added author and checkstyle related errors resolved

added the testcases to validate that useSslProtocol and useSslProtocol(algorithm) methods are never invoked

refactored the visibility modifiers

updated the method name to use camel case

instance checking with X509TrustManager removed and methods cleaned

skipServerCertificateValidation flag added

added comments

\* Polishing and reinstate `sslAlgorithmSet` logic

---

---

**\*\*Cherry-pick to 1.6.x and 1.7.x but with the `skipServerCertificateValidation = true` by default to keep the current behavior\*\***

(b). Number of affected files: 2

(c). Number of affected directories: 2

(d). Number of deleted lines (including comments and blank lines): 4

(e). Number of added lines (including comments and blank lines): 108

(f). Number of deleted lines (including comments and blank lines): 4

(g). Number of added lines (including comments and blank lines): 74

(h). Number of days between the current fixing commit and the previous commit:

spring-  
rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnectionFactoryBean.java: 18.15

spring-  
rabbit/src/test/java/org/springframework/amqp/rabbit/connection/SSLConnectionTests.java:  
71.99

The average number of days between two commits: 45.07

(i). Number of times each file has been modified:

spring-  
rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnectionFactoryBean.java: 11

spring-  
rabbit/src/test/java/org/springframework/amqp/rabbit/connection/SSLConnectionTests.java:  
10

The average number of modified times: 10.50

(j). Developers who have modified each file:

Developers	who	have	modified	file:	spring-
					rabbit/src/main/java/org/springframework/amqp/rabbit/connection/RabbitConnectionFactoryBean.java

harrydevnull

Heath Abelson

Arnaud Cogoluègnes

---

---

```

    Gary Russell

    Developers      who      have      modified      file:      spring-
rabbit/src/test/java/org/springframework/amqp/rabbit/connection/SSLConnectionTests.java

    harrydevnull

    Heath Abelson

    Artem Bilan

    Gary Russell

    The average number of developers modified each file: 4.00

(k). Number of commits each developer has made:

    Arnaud Cogoluègnes: 1  This is an new developer.

    Artem Bilan: 92  This is an new developer.

    Gary Russell: 461  This is an experienced developer.

    Heath Abelson: 1  This is an new developer.

    harrydevnull: 1  This is an new developer.

    Total number of developers modified the affected files: 5

```

---

### 3.2 Case 2

- CVE-ID: CVE-2018-11797
- Link: <https://github.com/apache/pdfbox>
- Fixing Commit: 4fa98533358c106522cd1bfe4cd9be2532af852

Output of case 2:

---

```

Analyses      of      repo:      https://github.com/apache/pdfbox,      commit:
4fa98533358c106522cd1bfe4cd9be2532af852

```

Current fix commit:

HEAD is now at 4fa985333 PDFBOX-4283: treat invalid elements as 0

(a). Commit message and title:

PDFBOX-4283: treat invalid elements as 0

git-svn-id: https://svn.apache.org/repos/asf/pdfbox/trunk@1837439 13f79535-47bb-0310-9956-ffa450edef68

(b). Number of affected files: 1

---



---

(c). Number of affected directories: 1

(d). Number of deleted lines (including comments and blank lines): 2

(e). Number of added lines (including comments and blank lines): 4

(f). Number of deleted lines (including comments and blank lines): 2

(g). Number of added lines (including comments and blank lines): 4

(h). Number of days between the current fixing commit and the previous commit:

pdfbox/src/main/java/org/apache/pdfbox/cos/COSArray.java: 197.07

The average number of days between two commits: 197.07

(i). Number of times each file has been modified:

pdfbox/src/main/java/org/apache/pdfbox/cos/COSArray.java: 35

The average number of modified times: 35.00

(j). Developers who have modified each file:

Developers	who	have	modified	file:
pdfbox/src/main/java/org/apache/pdfbox/cos/COSArray.java	John Hewson			
	Jeremias Maerki			
	Daniel Wilson			
	Andreas Lehmkuhler			
	Adam Nichols			
	Tilman Hausherr			
	Jukka Zitting			

The average number of developers modified each file: 7.00

(k). Number of commits each developer has made:

Adam Nichols: 46 This is an new developer.

Andreas Lehmkuhler: 1178 This is an experienced developer.

Jeremias Maerki: 36 This is an new developer.

John Hewson: 761 This is an experienced developer.

Jukka Zitting: 236 This is an experienced developer.

Tilman Hausherr: 3853 This is an experienced developer.

Total number of developers modified the affected files: 6

---

### 3.3 Case 3

- CVE-ID: CVE-2017-5647

- Link: <https://github.com/apache/tomcat80>
- Fixing Commit: ec10b8c785d1db91fe58946436f854dde04410fd

Output of case 3:

---

Analyses of repo: <https://github.com/apache/tomcat80>, commit: ec10b8c785d1db91fe58946436f854dde04410fd

Current fix commit:

HEAD is now at ec10b8c78 Improve sendfile handling when requests are pipelined.

(a). Commit message and title:

Improve sendfile handling when requests are pipelined.

git-svn-id: <https://svn.apache.org/repos/asf/tomcat/tc8.0.x/trunk@1788999> 13f79535-47bb-0310-9956-ffa450edef68

(b). Number of affected files: 9

(c). Number of affected directories: 3

(d). Number of deleted lines (including comments and blank lines): 36

(e). Number of added lines (including comments and blank lines): 136

(f). Number of deleted lines (including comments and blank lines): 30

(g). Number of added lines (including comments and blank lines): 94

(h). Number of days between the current fixing commit and the previous commit:

```
java/org/apache/coyote/AbstractProtocol.java: 146.19
java/org/apache/coyote/http11/Http11AprProcessor.java: 107.10
java/org/apache/coyote/http11/Http11Nio2Processor.java: 401.89
java/org/apache/coyote/http11/Http11NioProcessor.java: 401.89
java/org/apache/tomcat/util/net/AprEndpoint.java: 107.10
java/org/apache/tomcat/util/net/Nio2Endpoint.java: 336.03
java/org/apache/tomcat/util/net/NioEndpoint.java: 65.97
java/org/apache/tomcat/util/net/SendfileKeepAliveState.java: new added file 0
webapps/docs/changelog.xml: 1.13
```

The average number of days between two commits: 174.14

(i). Number of times each file has been modified:

```
java/org/apache/coyote/AbstractProtocol.java: 107
```

---

---

java/org/apache/coyote/http11/Http11AprProcessor.java: 177  
java/org/apache/coyote/http11/Http11Nio2Processor.java: 255  
java/org/apache/coyote/http11/Http11NioProcessor.java: 252  
java/org/apache/tomcat/util/net/AprEndpoint.java: 268  
java/org/apache/tomcat/util/net/Nio2Endpoint.java: 41  
java/org/apache/tomcat/util/net/NioEndpoint.java: 343  
java/org/apache/tomcat/util/net/SendfileKeepAliveState.java: 11  
webapps/docs/changelog.xml: 2632  
The average number of modified times: 454.00

(j). Developers who have modified each file:

Developers who have modified file: java/org/apache/coyote/AbstractProtocol.java

Filip Hanik  
Rainer Jung  
Mark Thomas  
Konstantin Kolinko  
Remy Maucherat  
Mladen Turk

Developers	who	have	modified	file:
------------	-----	------	----------	-------

java/org/apache/coyote/http11/Http11AprProcessor.java

Christopher Schultz  
Jean-Frederic Clere  
Violeta Georgieva Georgieva  
Filip Hanik  
Rainer Jung  
Mark Thomas  
Konstantin Kolinko  
Remy Maucherat  
Mladen Turk  
William Barker

Developers	who	have	modified	file:
------------	-----	------	----------	-------

java/org/apache/coyote/http11/Http11Nio2Processor.java

Filip Hanik  
Rainer Jung

---

---

Mark Thomas

Konstantin Kolinko

Remy Maucherat

Mladen Turk

William Barker

Developers                      who                      have                      modified                      file:

java/org/apache/coyote/http11/Http11NioProcessor.java

Christopher Schultz

Filip Hanik

Rainer Jung

Mark Thomas

Konstantin Kolinko

Remy Maucherat

Mladen Turk

William Barker

Developers who have modified file: java/org/apache/tomcat/util/net/AprEndpoint.java

Christopher Schultz

Jean-Frederic Clere

Violeta Georgieva Georgieva

Filip Hanik

Rainer Jung

Mladen Turk

Mark Thomas

Konstantin Kolinko

Remy Maucherat

Felix Schumacher

Costin Manolache

Developers who have modified file: java/org/apache/tomcat/util/net/Nio2Endpoint.java

Christopher Schultz

Rainer Jung

Mark Thomas

Konstantin Kolinko

Remy Maucherat

---

---

Developers who have modified file: java/org/apache/tomcat/util/net/NioEndpoint.java

Christopher Schultz

Peter Rossbach

Filip Hanik

Mladen Turk

Mark Thomas

Tim Whittington

Remy Maucherat

Konstantin Kolinko

Costin Manolache

Developers	who	have	modified	file:
------------	-----	------	----------	-------

java/org/apache/tomcat/util/net/SendfileKeepAliveState.java

Filip Hanik

Mark Thomas

Konstantin Kolinko

Remy Maucherat

Mladen Turk

Developers who have modified file: webapps/docs/changelog.xml

Coty Sutherland

Mark Thomas

Konstantin Kolinko

Remy Maucherat

Christopher Schultz

Peter Rossbach

Jeremy Boynes

kfujino

Keiichi Fujino

Timothy A. Funk

Jean-Frederic Clere

Yoav Shapira

Violeta Georgieva Georgieva

Konstantin Preißer

Filip Hanik

---

---

Tim Whittington

Mladen Turk

Felix Schumacher

Sylvain Laurent

Rainer Jung

William Barker

The average number of developers modified each file: 9.11

(k). Number of commits each developer has made:

Christopher Schultz: 81 This is an new developer.

Costin Manolache: 75 This is an new developer.

Coty Sutherland: 3 This is an new developer.

Felix Schumacher: 99 This is an new developer.

Filip Hanik: 1097 This is an experienced developer.

Jean-Frederic Clere: 81 This is an new developer.

Jeremy Boynes: 72 This is an new developer.

Keiichi Fujino: 365 This is an experienced developer.

Konstantin Kolinko: 1569 This is an experienced developer.

Konstantin Preißer: 71 This is an new developer.

Mark Thomas: 10186 This is an experienced developer.

Mladen Turk: 129 This is an experienced developer.

Peter Rossbach: 93 This is an new developer.

Rainer Jung: 434 This is an experienced developer.

Remy Maucherat: 641 This is an experienced developer.

Sylvain Laurent: 30 This is an new developer.

Tim Whittington: 23 This is an new developer.

Timothy A. Funk: 61 This is an new developer.

Violeta Georgieva Georgieva: 283 This is an experienced developer.

William Barker: 44 This is an new developer.

Yoav Shapira: 8 This is an new developer.

kfujino: 1 This is an new developer.

Total number of developers modified the affected files: 22

---