

Stack Overflow: the Vulnerability Market Place

First TC 2019

Danny Grander

danny@snyk.io

About me

- Danny Grander, [@grander](#) on Twitter
- Co-founder & Security Research at Snyk
- CTF player ([@pastenctf](#) team)







<> FilePath.java

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

`<>` **FilePath.java**

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsoluteFile();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

`<>` **FilePath.java**

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsoluteFile();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

`<>` **FilePath.java**

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```


FilePath.java

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

`<>` FilePath.java

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

 **FilePath.java**

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

FilePath.java

```
1  private void unzip(File dir, File zipFile) throws IOException {
2      dir = dir.getAbsolutePath();
3      ZipFile zip = new ZipFile(zipFile);
4
5      Enumeration<ZipEntry> entries = zip.getEntries();
6
7      while (entries.hasMoreElements()) {
8          ZipEntry e = entries.nextElement();
9          File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
19 }
```

backup.zip

Date	Time	Attr	Size	Compressed	Name
2018-07-17	16:56:14	78	78	backup.txt
2018-07-17	16:56:14		78	78	1 files

`unzip("/tmp/extracted/", "backup.zip")`

`/tmp/extracted/backup.txt`

evil_backup.zip

Date	Time	Attr	Size	Compressed	Name
2018-07-17	16:56:14	78	78	backup.txt
2018-07-17	16:56:14	237	120	../../../../../../../../home/root/.ssh/authorized_keys
2018-07-17	16:56:14		315	198	2 files

evil_backup.zip

Date	Time	Attr	Size	Compressed	Name
2018-07-17	16:56:14	78	78	backup.txt
2018-07-17	16:56:14	237	120	../../../../../../../../home/root/.ssh/authorized_keys
2018-07-17	16:56:14		315	198	2 files

`/tmp/extracted/../../../../../../../../home/root/.ssh/
authorized_keys`



`/home/root/.ssh/authorized_keys`

Impact

- Gain code execution by overwriting
 - Code files
 - Initialization scripts
 - Configuration files
 - Credentials and SSH Keys

Not only **ZIP!**

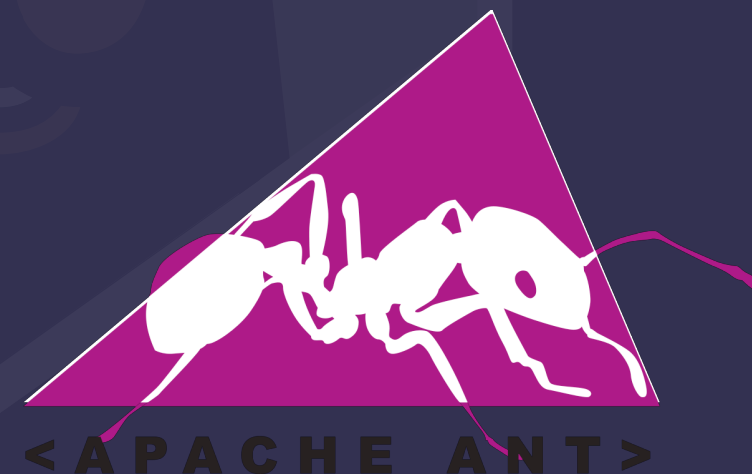
Archiver vs Compressor

Extension	Archiver	Compressor
zip (jar, war, apk)	yes	yes
7z	yes	yes
tar	yes	no
cpio	yes	no
.a / .ar	yes	no
gzip	no	yes





THE
APACHE[®]
SOFTWARE FOUNDATION





eclipse



sonarqube



Many more...

Archiving libraries



Affected archiving libraries



- unzipper
- adm-zip



Affected archiving libraries



- unzipper
- adm-zip

.NET

- DotNetZip
- SharpCompress
- SharpZipLib

Affected archiving libraries



- unzipper
- adm-zip

.NET

- DotNetZip
- SharpCompress
- SharpZipLib



- Oracle std. lib
- Apache commons-compress
- plexus-archiver
- zt-zip
- zip4j

Affected archiving libraries



- unzipper
- adm-zip

.NET

- DotNetZip
- SharpCompress
- SharpZipLib



- Oracle std. lib
- Apache commons-compress
- plexus-archiver
- zt-zip
- zip4j



- zip-ruby
- rubyzip
- zipruby

Affected archiving libraries



- unzipper
- adm-zip

.NET

- DotNetZip
- SharpCompress
- SharpZipLib



- Oracle std. lib
- Apache commons-compress
- plexus-archiver
- zt-zip
- zip4j



- zip-ruby
- rubyzip
- zipruby



- mholt/archiver
- cf/archiver

1991

Title : The Complete Guide to Hacking WWIV

Author : Inhuman

==Phrack Inc.==

Volume Three, Issue Thirty-four, File #5 of 11

```
***                                     ***
***                                     ***
*** The Complete Guide                 ***
*** to Hacking WWIV                   ***
***                                     ***
***      by Inhuman                    ***
***      September 1991                ***
***                                     ***
***                                     ***
```

WWIV is one of the most popular BBS programs in the country. With thousands of boards in WWIVnet and hundreds in the spinoff WWIVlink, there is a lot of support and community. The nice thing about WWIV is that it is very easy to set up. This makes it popular among the younger crowd of sysops who can't comprehend the complexities of fossil drivers and batch files. In this

Is EVERYTHING Vulnerable?





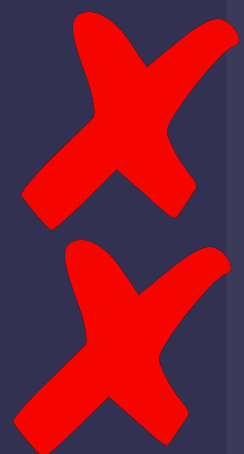
```
1  f = zipfile.ZipFile(zip_file, 'r')
2  f.extractall(destination_folder)
3  f.close()
```

.NET

```
1 ZipFile.ExtractToDirectory(zipPath, extractPath);
```



```
ZipFile.unzip(zipPath, extractPath);
```



Java Standard Library

Apache Commons Compress



```
3     ZipFile zip = new ZipFile(zipFile);
4
5     Enumeration<ZipEntry> entries = zip.getEntries();
6
7     while (entries.hasMoreElements()) {
8         ZipEntry e = entries.nextElement();
9         File f = new File(dir, e.getName());
10
11         if (e.isDirectory()) {
12             mkdirs(f);
13         } else {
14             try (InputStream input = zip.getInputStream(e)) {
15                 IOUtils.copy(input, writing(f));
16             }
17         }
18     }
```

Google

Google Search

I'm Feeling Lucky

Google offered in: [العربية](#) [עברית](#)





- ✗ How to unzip files programmatically in Android?
- ✗ What is a good Java library to zip/unzip files?
- ✗ Java ZIP – how to unzip folder?
- ✗ How do I extract a tar file in Java?
- ✗ How to untar a TAR file using Apache Commons
- ✗ Utility to unzip an entire archive to a directory in java
- ✗ Unzip Archive with Groovy
- ✗ Simplest way to download and unzip files in Node.js cross-platform?
- ✗ unzip (zip, tar, tag.gz) files with ruby



stackoverflow



I'd like to do something like this in my program:

10

```
File zipFile = .....;
File destDir = .....;
ImaginaryZipUtility.unzipAllTo(zipFile, destdir);
```



I cannot possibly be the first to do this from a program. Where do I find a utility method like above? I tried to look at apache commons-io, but nothing there. So, where should I look?

4

java unzip

share improve this question

asked Aug 31 '10 at 19:50



eirikma

537 ● 4 ● 12

I added this as a feature request at Apache commons-compress: issues.apache.org/jira/browse/COMPRESS-118 – eirikma Aug 31 '10 at 21:11

5 We do have now 2011 and there isn't even a (common) 3rd party library to extract a ZIP in Java with a single call? WTF – Kutzi Oct 17 '11 at 14:28



stackoverflow



I'd like to do something like this in my program:

10

```
File zipFile = .....;
File destDir = .....;
ImaginaryZipUtility.unzipAllTo(zipFile, destdir);
```



I cannot possibly be the first to do this from a program. Where do I find a utility method like above? I tried to look at apache commons-io, but nothing there. So, where should I look?

4

java unzip

share improve this question

asked Aug 31 '10 at 19:50



eirikma

537 ● 4 ● 12

I added this as a feature request at Apache commons-compress: issues.apache.org/jira/browse/COMPRESS-118 – eirikma Aug 31 '10 at 21:11

5 We do have now 2011 and there isn't even a (common) 3rd party library to extract a ZIP in Java with a single call? WTF – Kutzi Oct 17 '11 at 14:28

Zip Slip



- Lack of standard, high level API for archive extraction in some ecosystems
- Many vulnerable code snippets all around being copy & pasted
- Dozen of affected archive extraction libraries
- Hundreds of vulnerable projects

How do we **Disclose**?

- Responsible Disclosure
- 60 day disclosure period
- Coordinate and help fixing

Zip Slip CVEs

- CVE-2018-1002203
- CVE-2018-1002204
- CVE-2018-1002200
- CVE-2018-1002201
- CVE-2018-1002202
- CVE-2018-1002205
- CVE-2018-1002208
- CVE-2018-1002209
- CVE-2018-11762
- ...
- CVE-2018-1002206
- CVE-2018-1002207
- CVE-2018-8008
- CVE-2018-8009
- CVE-2018-1261
- CVE-2018-1263
- CVE-2018-10886
- CVE-2018-12036
- ...

<https://github.com/snyk/zip-slip-vulnerability>

If you find a library or project that contains similar vulnerable code, we ask for your contribution to this repository to provide the community with the most up to date information about the Zip Slip vulnerability. To contribute, please refer to our [CONTRIBUTE.md](#) file.

Affected Libraries

Many of the following affected libraries exist because their ecosystems lack high level APIs providing the basic archive management capabilities. This results in vulnerable code being shared and reused. The following table contains the list of vulnerable libraries we found during private disclosure of Zip Slip which we aim to keep up to date, with community support, going forward as more vulnerable libraries are discovered. Some libraries that do not provide the high-level API often result in vulnerable implementations also, either through people copying and pasting vulnerable private code, or writing their own vulnerable snippets.

Vendor	Product	Language	Confirmed vulnerable	Fixed Version	CVE	Fixed
npm library	unzipper	JavaScript	YES	0.8.13	CVE-2018-1002203	17/4/2018
npm library	adm-zip	JavaScript	YES	0.4.9	CVE-2018-1002204	23/4/2018
Java library	codehaus/plexus-archiver	Java	YES	3.6.0	CVE-2018-1002200	6/5/2018

Takeaways

Takeaway #1

Design simplified APIs with strong security defences implemented by default



```
f.extractall(destination_folder)
```

.NET

```
ZipFile.ExtractToDirectory(zipPath, extractPath);
```



```
ZipFile.unzip(zipPath, extractPath);
```



Takeaway #2

Don't be a "Full Stack Overflow Developer"



Takeaway #3

Proper archive handling

- Make sure you don't have a vulnerable **implementation** in your own code
- Make sure you're not using **libraries** with known vulnerabilities
- Don't trust archives that you not fully control

Affected archiving libraries



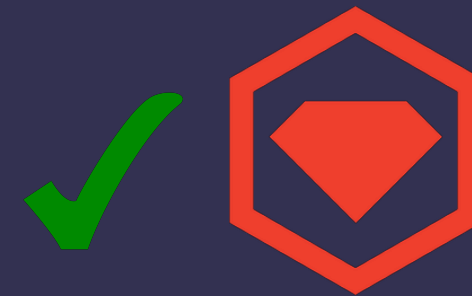
- unzipper
- adm-zip



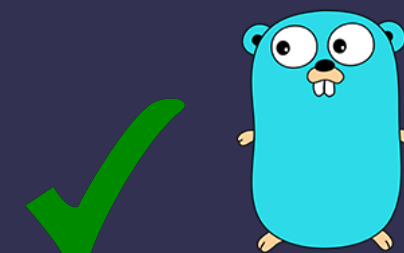
- DotNetZip
- SharpCompress
- SharpZipLib



- Oracle std. lib
- Apache commons-compress
- plexus-archiver
- zt-zip
- zip4j



- zip-ruby
- rubyzip
- zipruby



- mholt/archiver
- cf/archiver

Open Source Is Awesome

Please Enjoy Responsibly

Questions?

Danny Grander, Snyk
danny@snyk.io
@grander

Questions

- Zip Slip <https://snyk.io/research/zip-slip-vulnerability>
- Snyk blog <https://snyk.io/blog>
 - Zip Slip - behind the disclosure: <https://snyk.io/blog/behind-the-disclosure-the-zip-slip-vulnerability>
 - Attacking ftp clients <https://snyk.io/blog/attacking-an-ftp-client/>
- Snyk Vulnerability DB <https://snyk.io/vuln>
- BigQuery research
 - <https://medium.com/@hoffa>
 - <https://medium.com/@sAbakumoff>
- [The Secure Developer Podcast](#)



Other research

28 Sep 2017

Secure Coding Practices in Java: Challenges and Vulnerabilities

Na Meng, Stefan Nagy, Daphne Yao, Wenjie Zhuang, Gustavo Arango Argoty
Virginia Tech
Blacksburg, Virginia 24060
{nm8247,snagy2,danfeng,kaito,gustavo1}@vt.edu

ABSTRACT

Java platform and third-party libraries provide various security features to facilitate secure coding. However, misusing these features can cost tremendous time and effort of developers or cause security vulnerabilities in software. Prior research was focused on the misuse of cryptography and SSL APIs, but did not explore the key fundamental research question: what are the biggest challenges and vulnerabilities in secure coding practices? In this paper, we conducted a comprehensive empirical study on StackOverflow posts to understand developers' concerns on Java secure coding, their programming obstacles, and potential vulnerabilities in their code.

We observed that developers have shifted their effort to the usage of authentication and authorization features provided by Spring

1 INTRODUCTION

Java platform and third-party libraries or frameworks (e.g., BouncyCastle [7] and Spring Security [53]) provide various features to facilitate secure coding. However, misusing these libraries and frameworks not only costs excessive debugging effort of developers, but also leads to security vulnerabilities in software [13, 63, 95, 96]. For example, Veracode identified software errors in the handling of user credentials, including hard-coded password and plaintext passwords in configuration files [63]. These errors can enable attackers to bypass access controls.

Prior research mainly focused on the misuse of cryptography and SSL APIs that causes security vulnerabilities [78, 80, 83, 86]. Specifically, Lazar et al. manually examined 269 published cryptographic



“

In one instance, after accepting the vulnerable solution, an asker commented “Adding `csrf().disable()` solved the issue!!! I have no idea why it was enabled by default.”



“

3 out of 6 hashing-relevant posts accepted vulnerable solutions as correct answers, indicating that developers were unaware of best secure programming practices. Incorrect security information may propagate among Stack Overflow users and negatively influence software development.

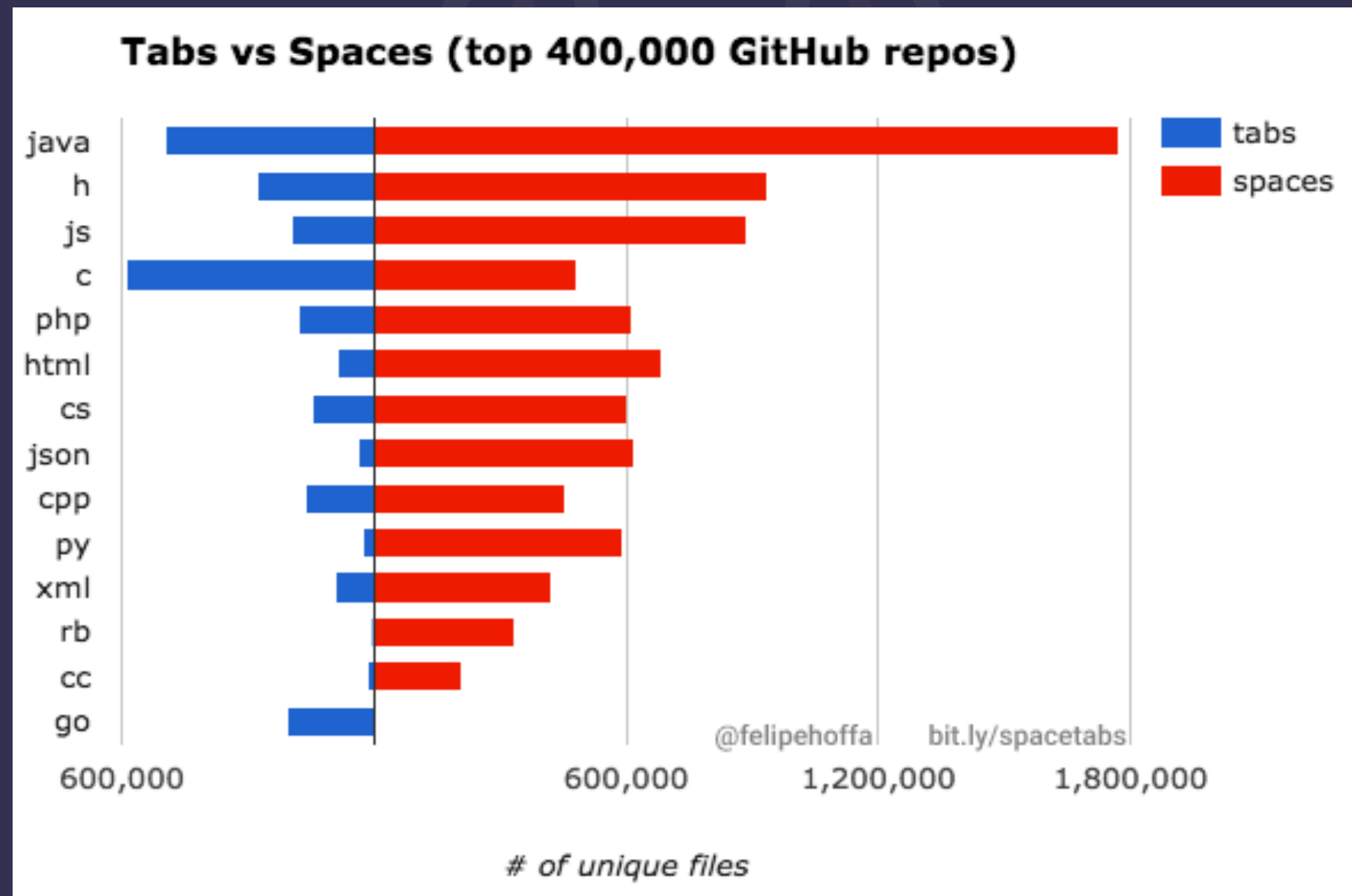


Google BigQuery



stackoverflow

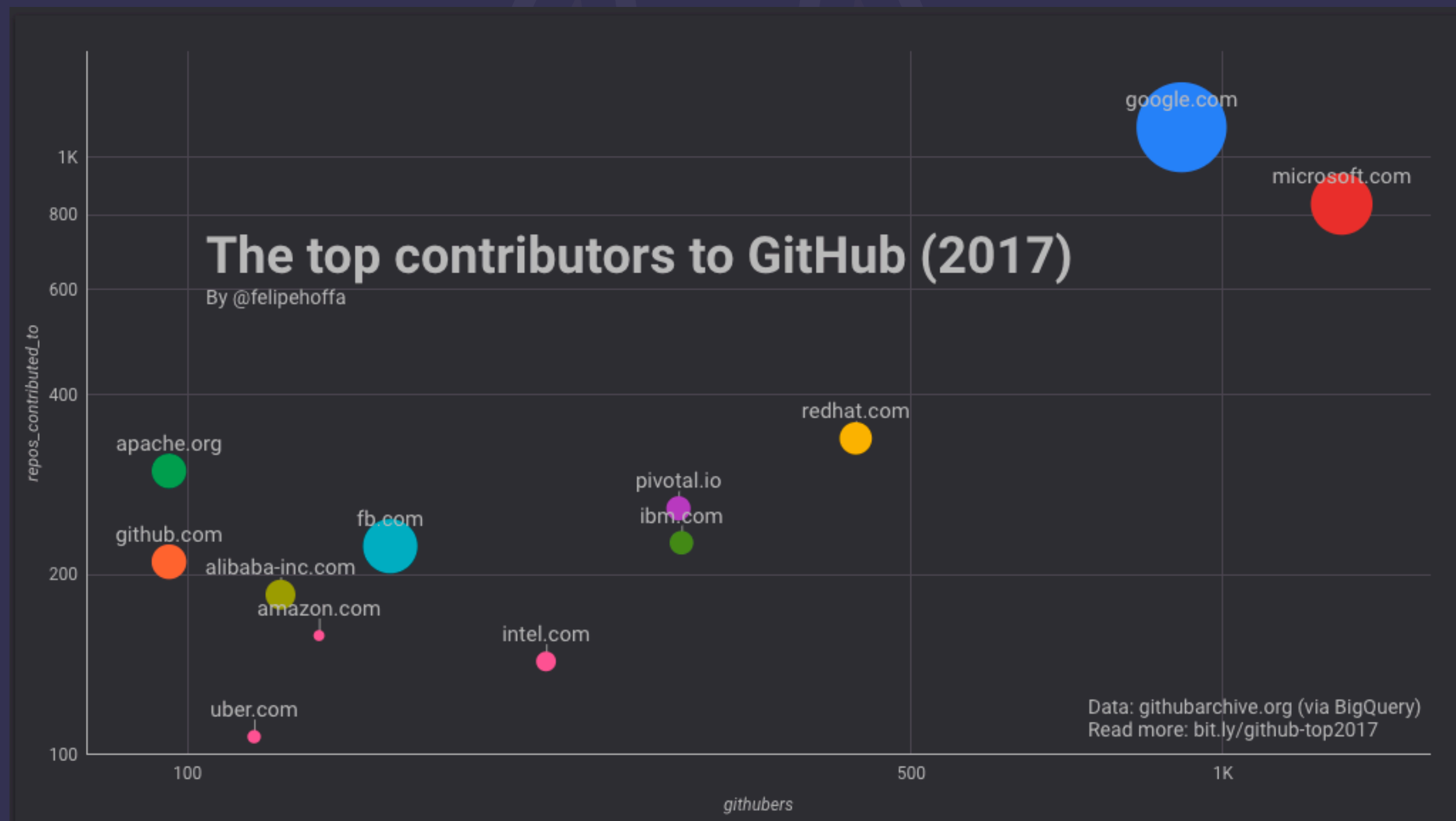
Spaces or Tabs?



F-bombs in commit messages

Results	Explanation	Download as CSV	Download as JSON
Row	commit	repo_name	
11	41cfe81e0171605e37d1e898ef58b8b98207c854	peck94/linode	fuck this
12	7b9c39cc4e3caf104e26510c4651621b2ea78c44	DragonLz/CYXBS_Android_V2.0	fuck ignore api.java
13	45e65c733dda2d69bc7a0be101f3cc0e73867e9b	djeik/fuckdown2	resolve warnings; throw away some old code; tie in the parser The program
14	a66be71489760a1ad4828528973dfd8a8cf176ec	item4/ugoirra	fuck, I do not use httpretty
15	b6e3b8bbe0664fa07e84bd7fb24f928566794cf	everyoneselectronic/anchor-cms	fuck I think I broke something
16	6e8733eaeb1543e288f5308b31f33c4c1b3f0d3c	UltrosBot/Ultros	A bunch of shit that got cocked up by a merge. I think I fixed it. Apparently
17	ee2dbbdf930b2c9f6fc41f294daf7c08e46dff50	m85091081/hakurei	fuck u google cdn
18	364e281d27d0ab0519952375a3511b54970d3de7	UltrosBot/Ultros	anusdickfuckcockpep8
19	0c5d56804c2a9109a69afcc1815d60cf460b3067	MarkusHackspacher/unknown-horizons	Big fucking update to anims concerning new directory structure. git-svn-id:
20	1f96817037b980caeb238fb621dcdc3dcc54aebf	sbryant/arrakis-hubot	Add some fucking class to #a Needs more Benedict Cumberbatch
21	69b42953e84962eddbc6719686f1c0bc768ce505	kianilannoye/Chronicals	fucking http
22	6707df15005148dbceabcffe204f6ed0d1d31a14	sonicyang/CNC430	Implemented Circle code which is fucking too fat
23	c3487e3d937741d5b6591e1252f5fb69b5ade7b8	ixmatus/recipes	Adding a fuckin' burger
24	28156134c3b2b84c9fb1295c7b18592b31afd420	plexinc/plex-media-player	Get codec info from PMS info if possible startCodecsLoading() is now call

Top open source contributors



Questions

- Zip Slip <https://snyk.io/research/zip-slip-vulnerability>
- Snyk blog <https://snyk.io/blog>
 - Zip Slip - behind the disclosure: <https://snyk.io/blog/behind-the-disclosure-the-zip-slip-vulnerability>
 - Attacking ftp clients <https://snyk.io/blog/attacking-an-ftp-client/>
- Snyk Vulnerability DB <https://snyk.io/vuln>
- BigQuery research
 - <https://medium.com/@hoffa>
 - <https://medium.com/@sAbakumoff>
- [The Secure Developer Podcast](#)

