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)
```java
class FilePath {
    private void unzip(File dir, File zipFile) throws IOException {
        dir = dir.getAbsoluteFile();
       ZipFile zip = new ZipFile(zipFile);
        Enumeration<ZipEntry> entries = zip.getEntries();
        while (entries.hasMoreElements()) {
            ZipEntry e = entries.nextElement();
            File f = new File(dir, e.getName());
            if (e.isDirectory()) {
                mkdirs(f);
            } else {
                try (InputStream input = zip.getInputStream(e)) {
                    IOUtils.copy(input, writing(f));
                }
            }
        }
    }
}
```
```java
class FilePath {
    private void unzip(File dir, File zipFile) throws IOException {
        File dir = dir.getAbsoluteFile();
        ZipFile zip = new ZipFile(zipFile);

        Enumeration<ZipEntry> entries = zip.getEntries();
        while (entries.hasMoreElements()) {
            ZipEntry e = entries.nextElement();
            File f = new File(dir, e.getName());

            if (e.isDirectory()) {
                mkdirs(f);
            } else {
                try (InputStream input = zip.getInputStream(e)) {
                    IOUtils.copy(input, writing(f));
                }
            }
        }
    }
}
```
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsoluteFile();
    ZipFile zip = new ZipFile(zipFile);
    Enumeration<ZipEntry> entries = zip.getEntries();
    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());
        if (e.isDirectory()) {
            mkdirs(f);
        } else {
            try (InputStream input = zip.getInputStream(e)) {
                IOUtils.copy(input, writing(f));
            }
        }
    }
}
```java
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsoluteFile();
    ZipFile zip = new ZipFile(zipFile);
    Enumeration<ZipEntry> entries = zip.getEntries();

    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());

        if (e.isDirectory()) {
            mkdirs(f);
        } else {
            try (InputStream input = zip.getInputStream(e)) {
                IOUtils.copy(input, writing(f));
            }
        }
    }
}
```
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsolutePath();
    ZipFile zip = new ZipFile(zipFile);
    Enumeration<ZipEntry> entries = zip.getEntries();
    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());
        // code for unzipping files
    }
}
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsolutePath();
    ZipFile zip = new ZipFile(zipFile);

    Enumeration<ZipEntry> entries = zip.getEntries();

    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());

        if (e.isDirectory()) {
            mkdirs(f);
        } else {
            try (InputStream input = zip.getInputStream(e)) {
                IOUtils.copy(input, writing(f));
            }
        }
    }
}
```java
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsoluteFile();
    ZipFile zip = new ZipFile(zipFile);

    Enumeration<ZipEntry> entries = zip.getEntries();

    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());

        if (e.isDirectory()) {
            makedirs(f);
        } else {
            try (InputStream input = zip.getInputStream(e)) {
                IOUtils.copy(input, writing(f));
            }
        }
    }
}
```
private void unzip(File dir, File zipFile) throws IOException {
    dir = dir.getAbsolutePath();
    ZipFile zip = new ZipFile(zipFile);

    Enumeration<ZipEntry> entries = zip.getEntries();

    while (entries.hasMoreElements()) {
        ZipEntry e = entries.nextElement();
        File f = new File(dir, e.getName());

        if (e.isDirectory()) {
            mkdirs(f);
        } else {
            try (InputStream input = zip.getInputStream(e)) {
                IOUtils.copy(input, writing(f));
            }
        }
    }
}
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Attr</th>
<th>Size</th>
<th>Compressed</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-07-17</td>
<td>16:56:14</td>
<td>......</td>
<td>78</td>
<td>78</td>
<td>backup.txt</td>
</tr>
<tr>
<td>2018-07-17</td>
<td>16:56:14</td>
<td></td>
<td>78</td>
<td>78</td>
<td>1 files</td>
</tr>
</tbody>
</table>
unzip("/tmp/extracted/", "backup.zip")
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Attr</th>
<th>Size</th>
<th>Compressed</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-07-17</td>
<td>16:56:14</td>
<td>..</td>
<td>78</td>
<td>78</td>
<td>backup.txt</td>
</tr>
<tr>
<td>2018-07-17</td>
<td>16:56:14</td>
<td>..</td>
<td>237</td>
<td>120</td>
<td>../../../../../../../home/root/.ssh/authorized_keys</td>
</tr>
<tr>
<td>2018-07-17</td>
<td>16:56:14</td>
<td></td>
<td>315</td>
<td>198</td>
<td>2 files</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Attr</td>
<td>Size</td>
<td>Compressed</td>
<td>Name</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>------</td>
<td>------</td>
<td>------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2018-07-17 16:56:14</td>
<td>.....</td>
<td>78</td>
<td>78</td>
<td>78</td>
<td>backup.txt</td>
</tr>
<tr>
<td>2018-07-17 16:56:14</td>
<td>.....</td>
<td>237</td>
<td>120</td>
<td>120</td>
<td>../../../../../home/root/.ssh/authorized_keys</td>
</tr>
<tr>
<td>2018-07-17 16:56:14</td>
<td></td>
<td>237</td>
<td>120</td>
<td>120</td>
<td>2 files</td>
</tr>
</tbody>
</table>
Impact

- Gain code execution by overwriting
  - Code files
  - Initialization scripts
  - Configuration files
  - Credentials and SSH Keys
Not only ZIP!
# Archiver vs Compressor

<table>
<thead>
<tr>
<th>Extension</th>
<th>Archiver</th>
<th>Compressor</th>
</tr>
</thead>
<tbody>
<tr>
<td>zip (jar, war, apk)</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>7z</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>tar</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>cpio</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>.a / .ar</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>gzip</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
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

Java
- Oracle std. lib
- Apache commons-compress
- plexus-archiver
- zt-zip
- zip4j
Affected archiving libraries

- unzipperr
- adm-zip

.NET
- DotNetZip
- SharpCompress
- SharpZipLib

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

- zip-ruby
- rubyzip
- zipruby
Affected archiving libraries

- npm
  - unzipper
  - adm-zip

- .NET
  - DotNetZip
  - SharpCompress
  - SharpZipLib

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

- Ruby
  - zip-ruby
  - rubyzip
  - zipruby

- Go
  - mholt/archiver
  - cf/archiver
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?

WHY HASN'T ANYONE ELSE SEEN THIS BEFORE?
1. `f = zipfile.ZipFile(zip_file, 'r')`
2. `f.extractall(destination_folder)`
3. `f.close()`
1. `ZipFile.ExtractToDirectory(zipPath, extractPath);`
ZipFile.unzip(zipPath, extractPath);
Java Standard Library

Apache Commons Compress
ZipFile zip = new ZipFile(zipFile);

Enumeration<ZipEntry> entries = zip.getEntries();

while (entries.hasMoreElements()) {
    ZipEntry e = entries.nextElement();
    File f = new File(dir, e.getName());

    if (e.isDirectory()) {
        mkdirs(f);
    } else {
        try (InputStream input = zip.getInputStream(e)) {
            IOUtils.copy(input, writing(f));
        }
    }
}
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
I'd like to do something like this in my program:

```java
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?

[Java and Unzip tags]

asked Aug 31 '10 at 19:50 by eirikma

I added this as a feature request at Apache commons-compress: issues.apache.org/jira/browse/CMPRESS-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
I'd like to do something like this in my program:

```java
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?

JAVA UNZIP

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

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
<table>
<thead>
<tr>
<th>CVE-2018-1002203</th>
<th>CVE-2018-1002206</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2018-1002204</td>
<td>CVE-2018-1002207</td>
</tr>
<tr>
<td>CVE-2018-1002201</td>
<td>CVE-2018-8009</td>
</tr>
<tr>
<td>CVE-2018-1002202</td>
<td>CVE-2018-1261</td>
</tr>
<tr>
<td>CVE-2018-1002205</td>
<td>CVE-2018-1263</td>
</tr>
<tr>
<td>CVE-2018-1002208</td>
<td>CVE-2018-10886</td>
</tr>
<tr>
<td>CVE-2018-1002209</td>
<td>CVE-2018-12036</td>
</tr>
<tr>
<td>CVE-2018-11762</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
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 CONTRIBUTION.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.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Language</th>
<th>Confirmed vulnerable</th>
<th>Fixed Version</th>
<th>CVE</th>
<th>Fixed</th>
</tr>
</thead>
<tbody>
<tr>
<td>npm library</td>
<td>unzipping</td>
<td>JavaScript</td>
<td>YES</td>
<td>0.8.13</td>
<td>CVE-2018-1002203</td>
<td>17/4/2018</td>
</tr>
<tr>
<td>npm library</td>
<td>adm-zip</td>
<td>JavaScript</td>
<td>YES</td>
<td>0.4.9</td>
<td>CVE-2018-1002204</td>
<td>23/4/2018</td>
</tr>
<tr>
<td>Java library</td>
<td>codehaus/plexus-archiver</td>
<td>Java</td>
<td>YES</td>
<td>3.6.0</td>
<td>CVE-2018-1002200</td>
<td>6/5/2018</td>
</tr>
</tbody>
</table>
Takeaways
Takeaway #1
Design simplified APIs with strong security defences implemented by default.
f.extractall(destination_folder)

ZipFile.ExtractToDirectory(zipPath, extractPath);

ZipFile.unzip(zipPath, extractPath);
Takeaway #2

Don’t be a “Full Stack Overflow Developer”

https://christianheilmann.com/2015/07/17/the-full-stackoverflow-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

- **npm**
  - unzipper
  - adm-zip

- **.NET**
  - DotNetZip
  - SharpCompress
  - SharpZipLib

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

- **Ruby**
  - zip-ruby
  - rubyzip
  - zipruby

- **Dart**
  - mholt/archiver
  - cf/archiver

Open Source Is Awesome
Please Enjoy Responsibly

Questions?
Danny Grander, Snyk
danny@snyk.io
@grander
Questions

- Snyk blog https://snyk.io/blog
  - Zip Slip - behind the disclosure: https://snyk.io/blog/behind-the-disclosure-the-zip-slip-vulnerability
- Snyk Vulnerability DB https://snyk.io/vuln
- BigQuery research
  - https://medium.com/@hoffa
  - https://medium.com/@sAbakumoff
- The Secure Developer Podcast
Other research
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.”
Spaces or Tabs?

Tabs vs Spaces (top 400,000 GitHub repos)

- java
- h
- js
- c
- php
- html
- cs
- json
- cpp
- py
- xml
- rb
- cc
- go

# of unique files

https://medium.com/@hoffa/400-000-github-repositories-1-billion-files-14-terabytes-of-code-spaces-or-tabs-7cfe0b5dd7fd
## F-bombs in commit messages

A table showing a sample of commit messages containing F-bombs:

<table>
<thead>
<tr>
<th>Row</th>
<th>commit</th>
<th>repo_name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>41cfe81e0171605e37d1e898ef58b8b9207c854</td>
<td>peck94/linode</td>
<td>fuck this</td>
</tr>
<tr>
<td>12</td>
<td>7b9c39cc4e3cafc104e26510c4651621b2ea78c44</td>
<td>DragonLz/CXYBS_Android_V2.0</td>
<td>fuck ignore api.java</td>
</tr>
<tr>
<td>13</td>
<td>45e65c733da2d69bc7a0be101f3cc0e73867e9b</td>
<td>djeik/fuckdown2</td>
<td>resolve warnings; throw away some old code; tie in the parser The progr</td>
</tr>
<tr>
<td>14</td>
<td>a66be71489760a1ad4828528973dfd8a8cfl76ec</td>
<td>item4/ugoira</td>
<td>fuck, I do not use httpretty</td>
</tr>
<tr>
<td>15</td>
<td>b6e3b8bbbe0664af0e7e48b7fb2f4928566794c</td>
<td>everyoneelectronic/anchor-cms</td>
<td>fuck I think I broke something</td>
</tr>
<tr>
<td>16</td>
<td>6e8733eaeab1543e2885f5308b31f3c3c4c1b3f0d3c</td>
<td>UltrosBot/Ultros</td>
<td>A bunch of shit that got cocked up by a merge. I think I fixed it. Apparently</td>
</tr>
<tr>
<td>17</td>
<td>ee2d2bd930b2c966fc41294daf7c08e46ff50</td>
<td>m85091091/hakurei</td>
<td>fuck u google cdn</td>
</tr>
<tr>
<td>18</td>
<td>3642e581d27d0a6519952375a3511b54970d3de7</td>
<td>UltrosBot/Ultros</td>
<td>anusdickfuckcockpep8</td>
</tr>
<tr>
<td>19</td>
<td>05d566804c2a9109a69afec1815d60cf4603067</td>
<td>MarkusHackschaper/unknown-horizons</td>
<td>Big fucking update to anims concerning new directory structure. git-svn-id:</td>
</tr>
<tr>
<td>20</td>
<td>19f6817037b980caeb238fb621dcd3ccc54aefb</td>
<td>sbryant/arrakis-hubot</td>
<td>Add some fucking class to #a Needs more Benedict Cumberbatch</td>
</tr>
<tr>
<td>21</td>
<td>69b42953e84962eddef671996861f0c6766ce505</td>
<td>kianilannoys/Chronicals</td>
<td>fucking http</td>
</tr>
<tr>
<td>22</td>
<td>6707df15005148bceabcfeb20f4fed0dd131a14</td>
<td>soniycang/CNC430</td>
<td>Implemented Circle code which is fucking too fat</td>
</tr>
<tr>
<td>23</td>
<td>c3487e3d937741d56591e1252f5bf69b5ade7b8</td>
<td>ixmatus/recipes</td>
<td>Adding a fuckin’ burger</td>
</tr>
<tr>
<td>24</td>
<td>281561343c3b2884c9f1295c7b1659231af0d40</td>
<td>plexinc/plex-media-player</td>
<td>Get codec info from PMS info if possible startCodecsLoading() is now cali</td>
</tr>
</tbody>
</table>

Top open source contributors

https://medium.freecodecamp.org/the-top-contributors-to-github-2017-be98ab854e87
Questions

- Snyk blog [https://snyk.io/blog](https://snyk.io/blog)
- Snyk Vulnerability DB [https://snyk.io/vuln](https://snyk.io/vuln)
- BigQuery research
  - [https://medium.com/@hoffa](https://medium.com/@hoffa)
  - [https://medium.com/@sAbakumoff](https://medium.com/@sAbakumoff)
- The Secure Developer Podcast