It’s easier to Pirate with an Atlas...sian

This guide will introduce running Jira locally and setting up an IntelliJ debugger to identify code components and *hypothetically, for educational purposes* pirate Jira.

> ⚠️ Disclaimer: This guide is for educational purposes only, it is not an endorsement or instruction to pirate software, including Atlassian products like Jira. Software piracy is illegal and unethical, violating intellectual property rights and leading to legal consequences.
>
> I am not a lawyer etc etc.

Part 1: Set-up Jira

1. **Download Jira:** https://www.atlassian.com/software/jira/download/data-center & unzip it
2. **Install atlas plugin** (this is how we will run Jira locally)
    a. https://developer.atlassian.com/server/framework/atlassian-sdk/install-the-atlassian-sdk-on-a-linux-or-mac-system/
    b. https://developer.atlassian.com/server/framework/atlassian-sdk/atlas-run-standalone/

3. **Run Jira** (make sure to change the version to match what you downloaded, and take note of the debug port)
    a. You should check nothing is running on the port Jira is about to start up on (`1990` below), and the debug port (`8983` below)

```
atlas-run-standalone --product jira --version 9.4.0 \
--jvmargs "-Xdebug \
-Datlassian.dev.mode=false \
-Datlassian.disable.caches=false \
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address
-Xmx3g -Xms512m \
-XX:MaxMetaspaceSize=512m \
-Datlassian.webresource.disable.minification=true \
-Dsynchrony.proxy.enabled=false" \
--server localhost -p 1990
```

4. Login with the provided “admin” “admin” creds
5. Trigger a server error by changing the “F” to “G” in the free license provided at http://localhost:1990/jira/plugins/servlet/applications/versions-licenses

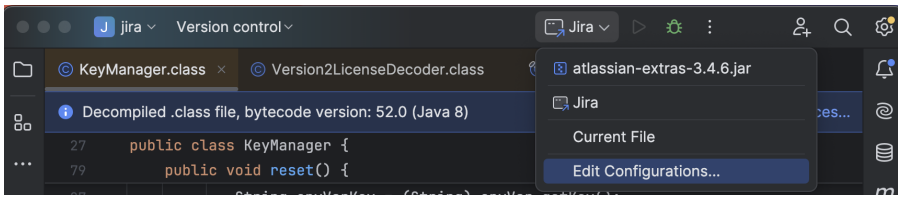
Yay Jira is working properly!!!

Part 2: Set-up IntelliJ

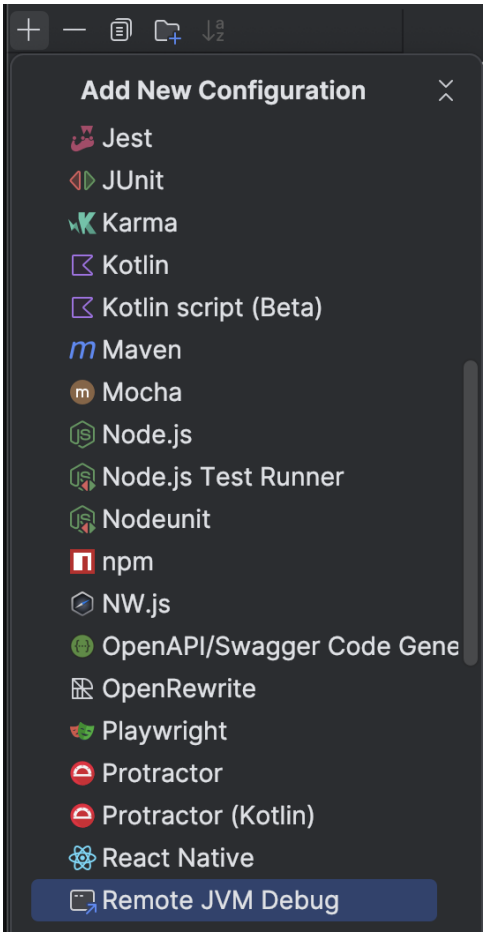
1. Test things
2. **Install IntelliJ**, open Jira as a “project” via the `effectivePom.xml` file
    a. For most people this will be located inside `~/amps-standalone-jira-9.4.0/target/jira` (or wherever you unzipped Jira)
3. Setup debugger

a. “Edit Configurations” → “+” → “Remote JVM Debug” → set the port to `8983` or whatever you selected above

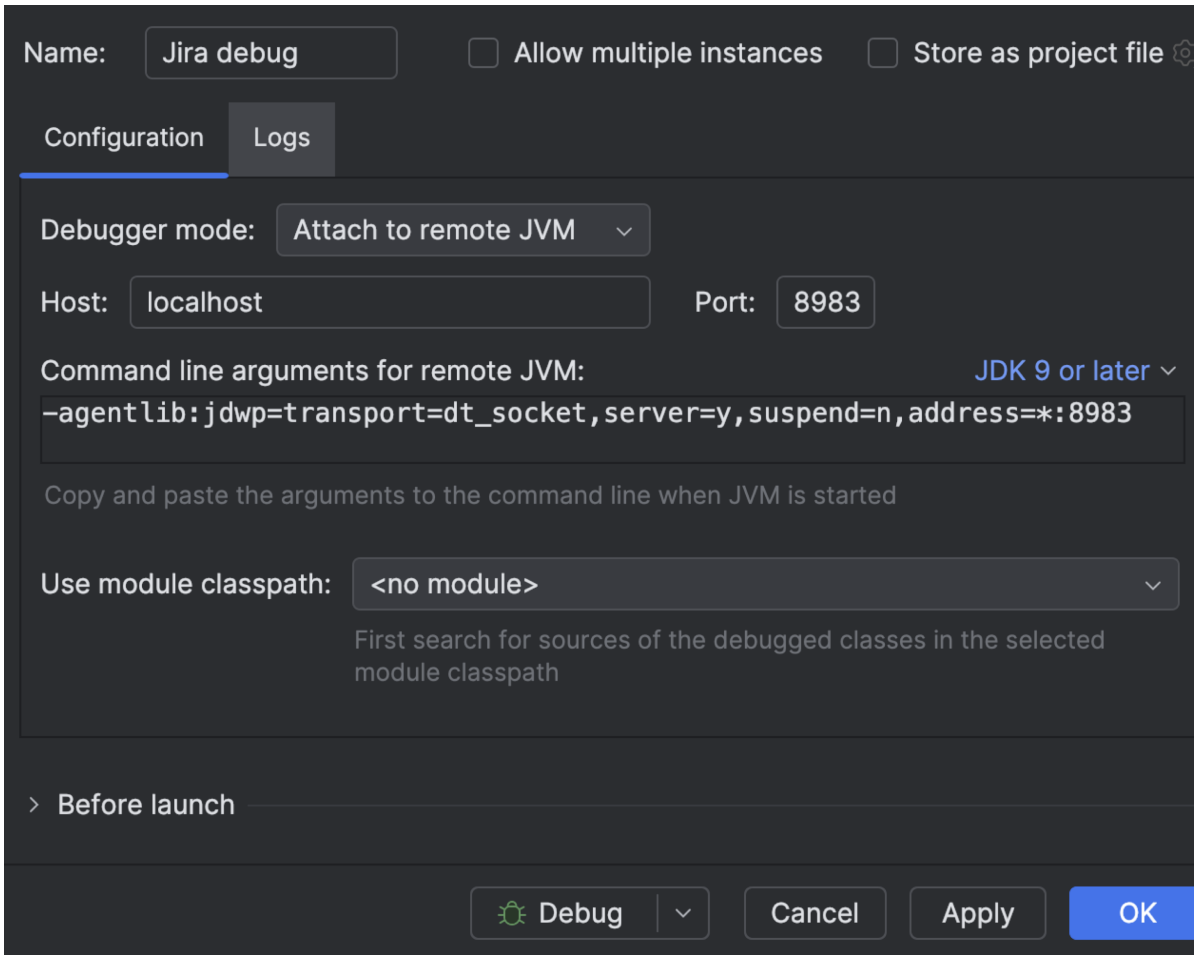
b. https://www.jetbrains.com/help/idea/debugging-code.html



Select Edit Configurations

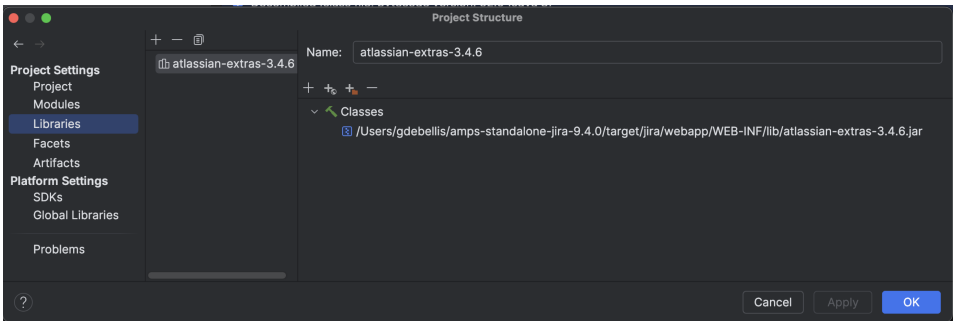


Select + then Remote JVM Debug



Ensure the port is set to 8983, everything else can stay the same

4. Locate the `atlassian-extras-3.4.6.jar` inside `~/amps-standalone-jira-9.4.0/target/jira` (I used finder file search)
5. Unzip the jar file with your favourite unzip tool (I used the terminal command `unzip -l atlassian-extras-3.4.6.jar`)
6. Find `Version2LicenseDecoder.class` inside the unzipped directory (I used `ls | grep Version2LicenseDecoder.class`)
7. Decompile the class file with IntelliJ
    a. Settings Icon → Project Structure → Libraries → + → Add `atlassian-extras-3.4.6.jar`
    b. IntelliJ might need some time to index the files before the next step



1. https://www.jetbrains.com/help/idea/decompiler.html
2. https://blog.jetbrains.com/idea/2020/03/java-bytecode-decompiler/

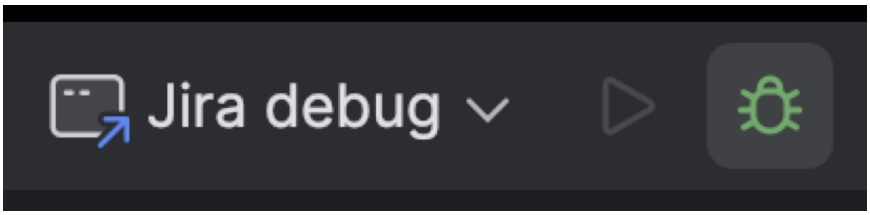
8. Search for `checkAndGetLicenseText` function within IntellIJ (I used CMD+F)
9. Set a break-point within this function (I set mine at the line below)

```
153     private byte[] checkAndGetLicenseText(String licenseContent) {
154         try {
155             byte[] decodedBytes = Base64.decodeBase64(licenseContent.getBytes(StandardCharsets.UTF_8));
156             ByteArrayInputStream in = new ByteArrayInputStream(decodedBytes);
157             DataInputStream dIn = new DataInputStream(in);
158             int textLength = dIn.readInt();
159             byte[] licenseText = new byte[textLength];
160             dIn.read(licenseText);
161             byte[] hash = new byte[dIn.available()];
162             dIn.read(hash);
163             String encodedLicenseText = new String(Base64.encodeBase64(licenseText), StandardCharsets.UTF_8);
164             String encodedHash = new String(Base64.encodeBase64(hash), StandardCharsets.UTF_8);
●               if (!KeyManager.getInstance().verify(encodedLicenseText, encodedHash, "LICENSE_STRING_KEY_V2")) {
166                 throw new LicenseException("Failed to verify the license.");
167             } else {
168                 return licenseText;
169             }
170         } catch (Exception var10) {
171             Exception e = var10;
172             throw new LicenseException(e);
173         }
174     }
```

Breakpoint set at verify()

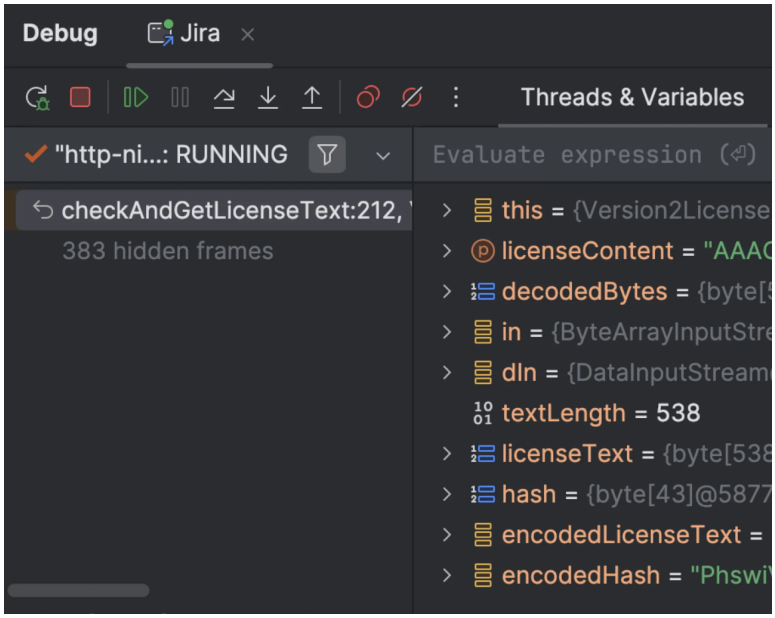
Part 3: Finding the code to edit

1. Run the debugger by hitting the green bug



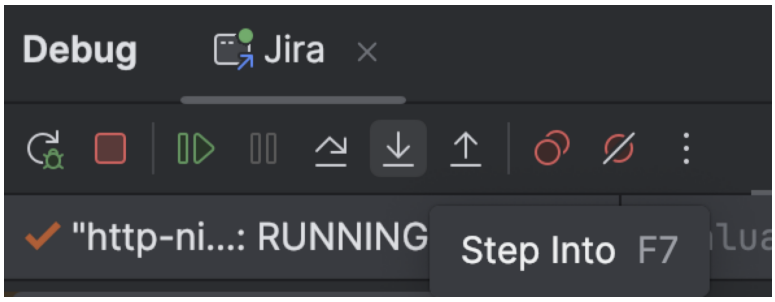
I called my debugger configuration “Jira debug”, you may have called yours something else

2. Recreate step 5 of part 1 (i.e. trigger the breakpoint). If the breakpoint gets set, you should see something like the following in the debug console)



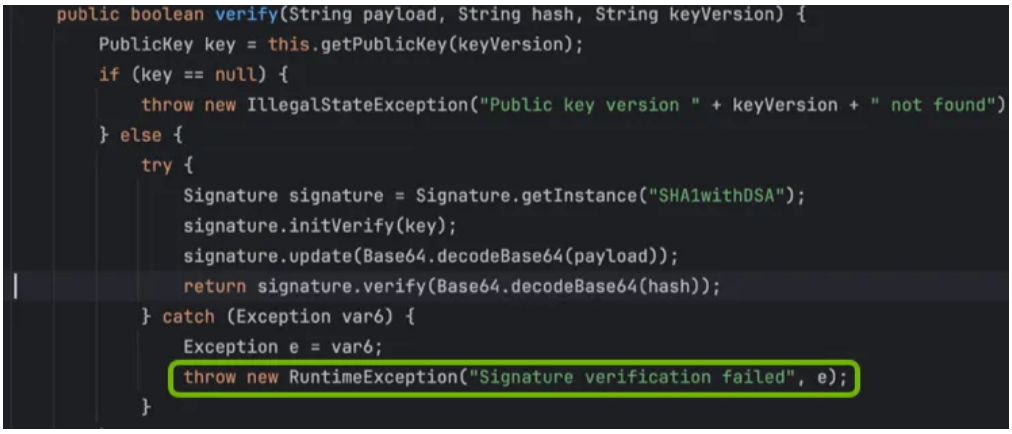
See the current variable state, this helps us figure out what input might be vulnerable!

3. “Step into” the `verify` function



There’s lots of debug options available, but we mostly want to step into functions, step over functions (go to the next line), or resume the program state

4. “Step over” lines within the verify function util the exception is raised, indicating where the license’s signature is verified



5. What if...

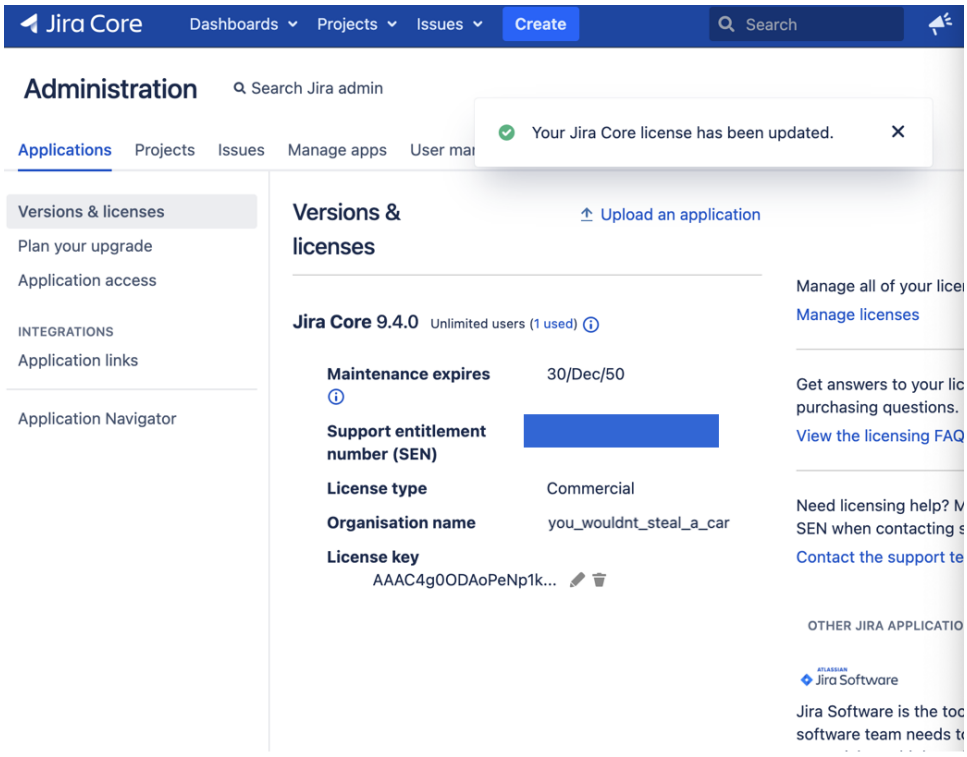
Part 4: Edit & Recompile code

✨ Left to the reader ✨

Part 5: Generate a fake license

✨ Left to the reader ✨

Part 6: 🤑



Conclusion

- Don’t be scared of big code repos!
- Hacking is the easiest when you use the tools made for developers to debug things
- Pirating is brat

Have fun hacking!

~ Giuliana

(I’m sorry I don’t use cool social medias, but you can message me on linkedIn and I’ll probably reply)