Tracking Vulnerable JARs

version 2.0 | 28-01-2013

Arun Babu Neelicattu | Red Hat Security Response | abn@redhat.com

Re-run

- The original
  - Ruxcon 2012
  - By David Jorm
  - SRT Veteran

- Some new stuff
  - Demos
  - How we leverage this internally

Link

http://www.ruxcon.org.au/speakers/#David Jorm

The What?

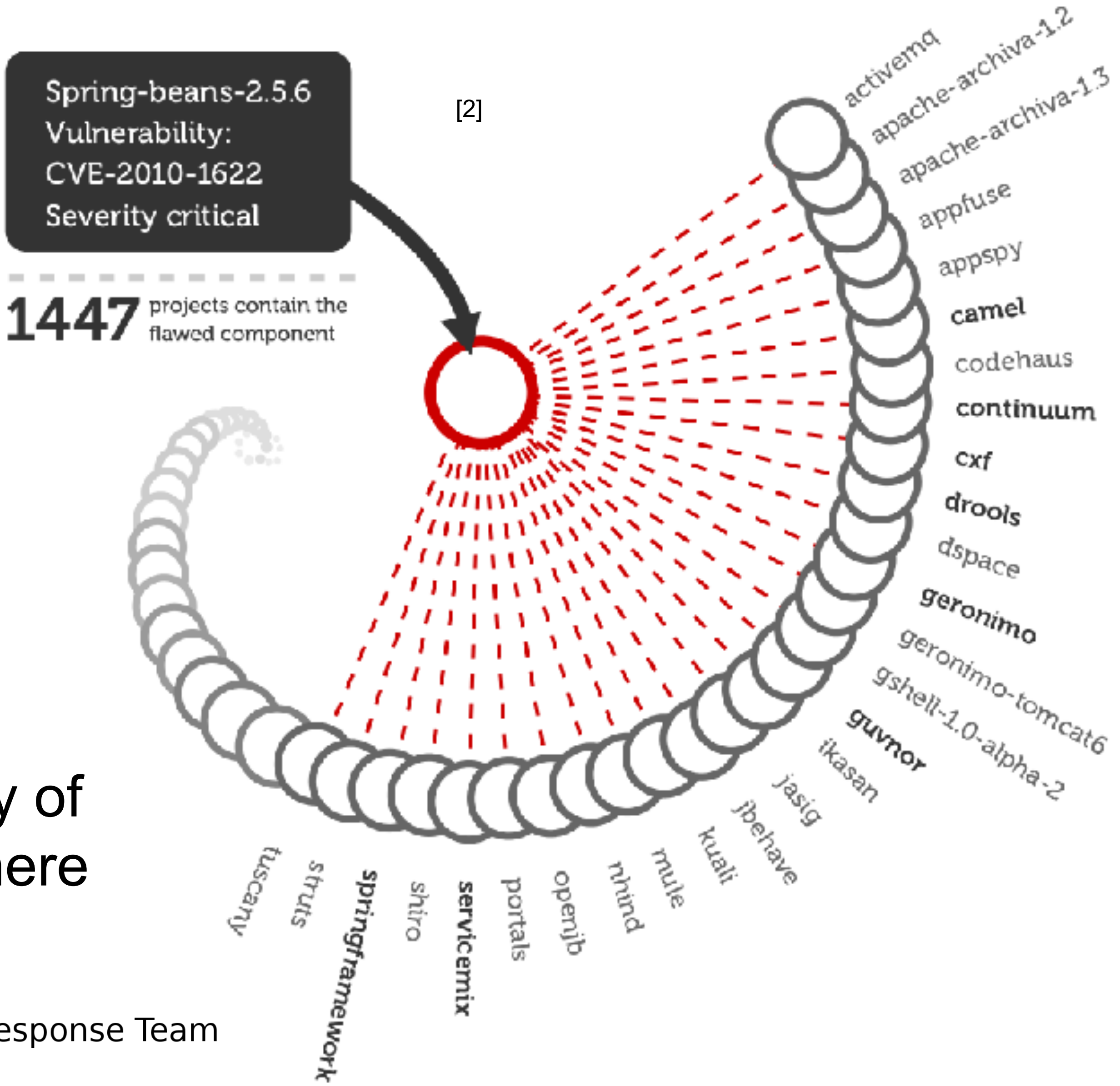
- The Why?
  - JAR (Security) Hell
  - A real world problem
  - JBoss product security

- The How?
  - Simple solution
  - Tools
  - How we do it
  - Demos

The Why? - JAR (Security) Hell

- A side effect of the "java" way
- Dependency management - By the app
- Dependency JARs are typically bundled
- Pulled by build tools like maven
- Drawn from public repositories
- Maven central repo - most “canonical” source of compiled JARs

The Why? - Real World Problem



[2]

There are plenty of examples out there

The Why?

Not sure if I am vulnerable
Or if they just forgot to bump the version number
quickmeme.com

The Why? - In numbers

- Aspect Security study in March 2012 [1]
  - Maven central repository library downloads
  - 29.8 million (26%) were for versions with known flaws
- Study recommendation to application developers
  - Provide tailored security policies that can be leveraged by the Java Security Manager to ensure limited impact of any exposure.
  - Enforce scans of dependencies against a known vulnerability DB.
  - Internalize and self manage Maven repositories to ensure absolute control of dependencies.

The Why? - JBoss Products

- Enterprise Application Platform 6.0.1
  - Modules
  - 742 Unique JARs bundled (source + build)
  - 1065 JARs (source + build)

- New vulnerability - Are we affected? How to patch?
  - Patch levels
  - Version
  - How it can be used?

The How? - A Solution

Enforce scans of dependencies against a known vulnerability DB - Aspect Security

- Feasible solution, simple enough
- But, no one has such a public database!

The How? - The Commercial Solution

- Sonatype “Insight App Health Check”
  - Includes source licensing and security checks
  - Available as GUI/maven plugin
  - Operates using remote service
  - $499 (per scan report?)

- Aspect Security “Contrast”
  - Identifies flaws in your own code
  - Maven plugin
  - Doesn't handle known flaws in dependencies yet
  - Free version, commercial $199-399/mo

The How? - A Solution

"Good news everyone!!"

- We are bringing together projects that could help!
- What we need
  - Central database
  - Catch flaws during development
  - Scan archives for flaws
- The projects:
  - Victi.ms
  - victims-enforcer maven plugin
  - jsnoop

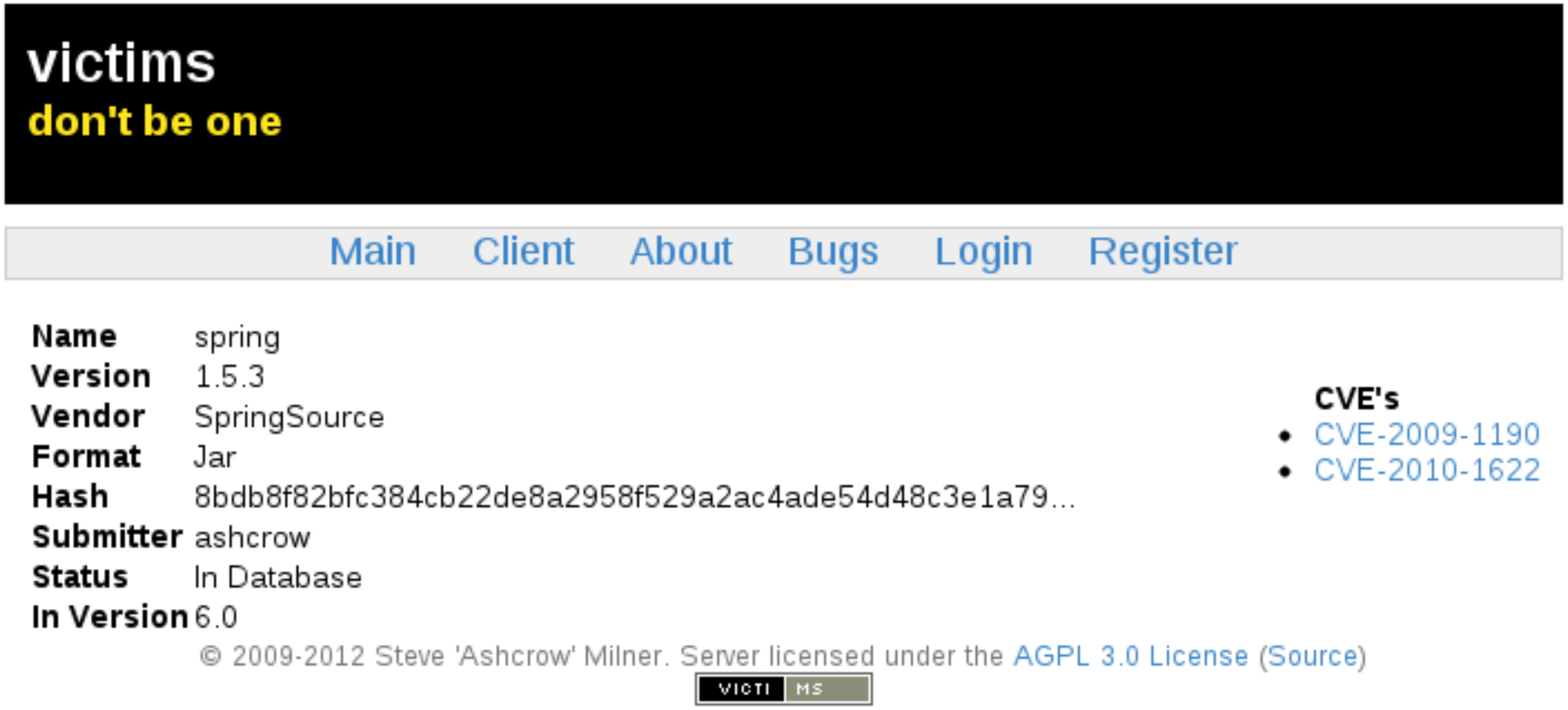
The How? - Victi.ms

- Victi.ms - Don't be one
- A central fingerprint database for vulnerable JARs
- By Steve Milner (Infosec Analyst)
- Resurrected last year
- Crowdsourced fingerprinting
- RH JARs added when a flaw is public
- Active development, version 2.0

Links

- Web: http://victi.ms
- Server: https://github.com/victims/victims-web
- Client: https://github.com/victims/victims-client
- Helper: https://github.com/victims/victims-hash

The How? - Victi.ms


victims

don't be one

Main Client About Bugs Login Register

Name	spring
Version	1.5.3
Vendor	SpringSource
Format	Jar
Hash	8bdb8f82bfc384cb22de8a2958f529a2ac4ade54d48c3e1a79...
Submitter	ashcrow
Status	In Database
In Version	6.0

CVE's

- CVE-2009-1190
- CVE-2010-1622

© 2009-2012 Steve 'Ashcrow' Milner. Server licensed under the AGPL 3.0 License (Source)

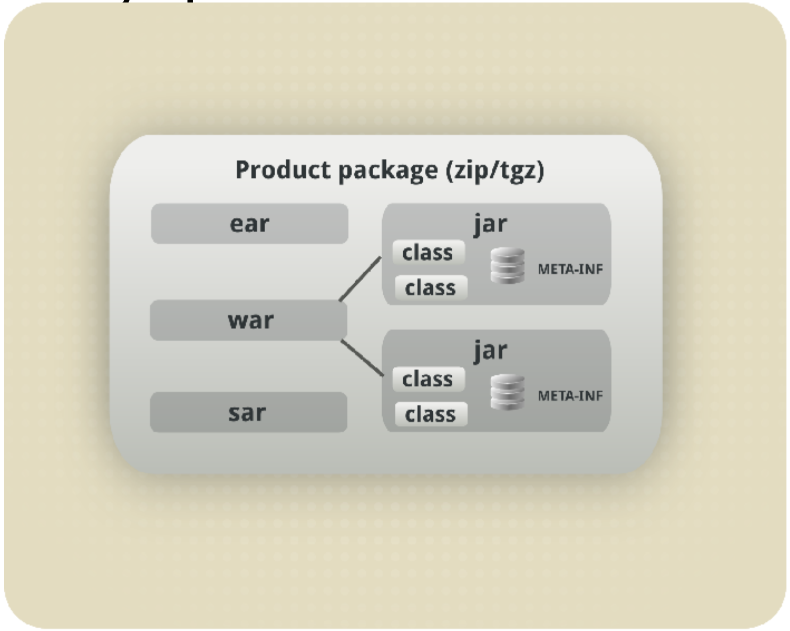
VICTI MS

The How? - Victims Enforcer

- Victims Enforcer (maven plugin)
- By Grant Murphy (Product Security Team)
- For the developers
- Uses victims DB to Fail/Warn builds
- Matches on SHA-512
- Detects issues at build-time
- Zero False Negatives
- https://github.com/victims/victims-enforcer
- Demo

The How? - JSnoop (New)

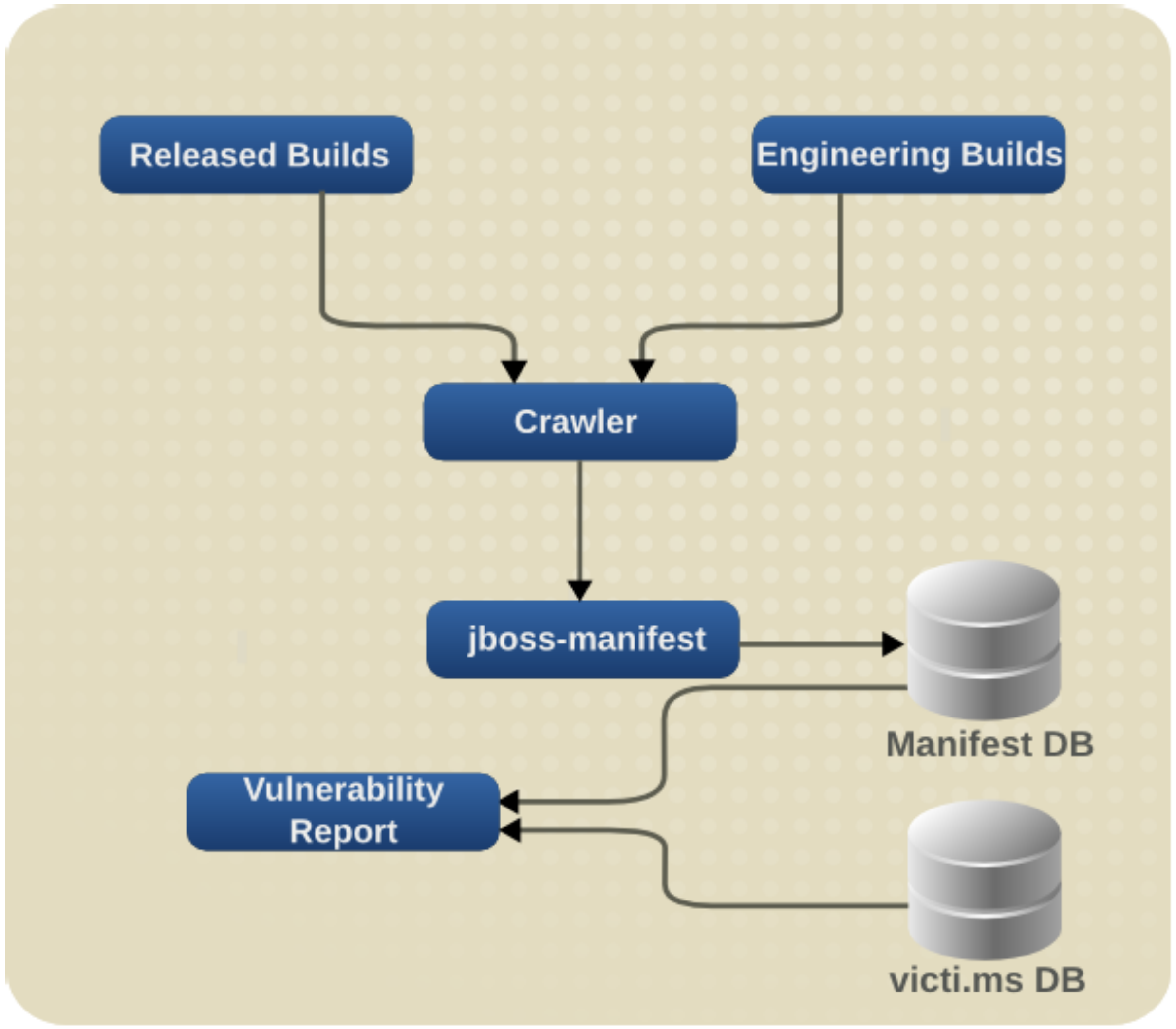
- Python3 module to handle recursive cataloging
- Initial implementation by me
- Active development
- Will get optional check agains Victi.ms
- Able to process entirely in-memory quick
- https://github.com/abn/jsnoop
- Demo



The How? - Manifest Tool

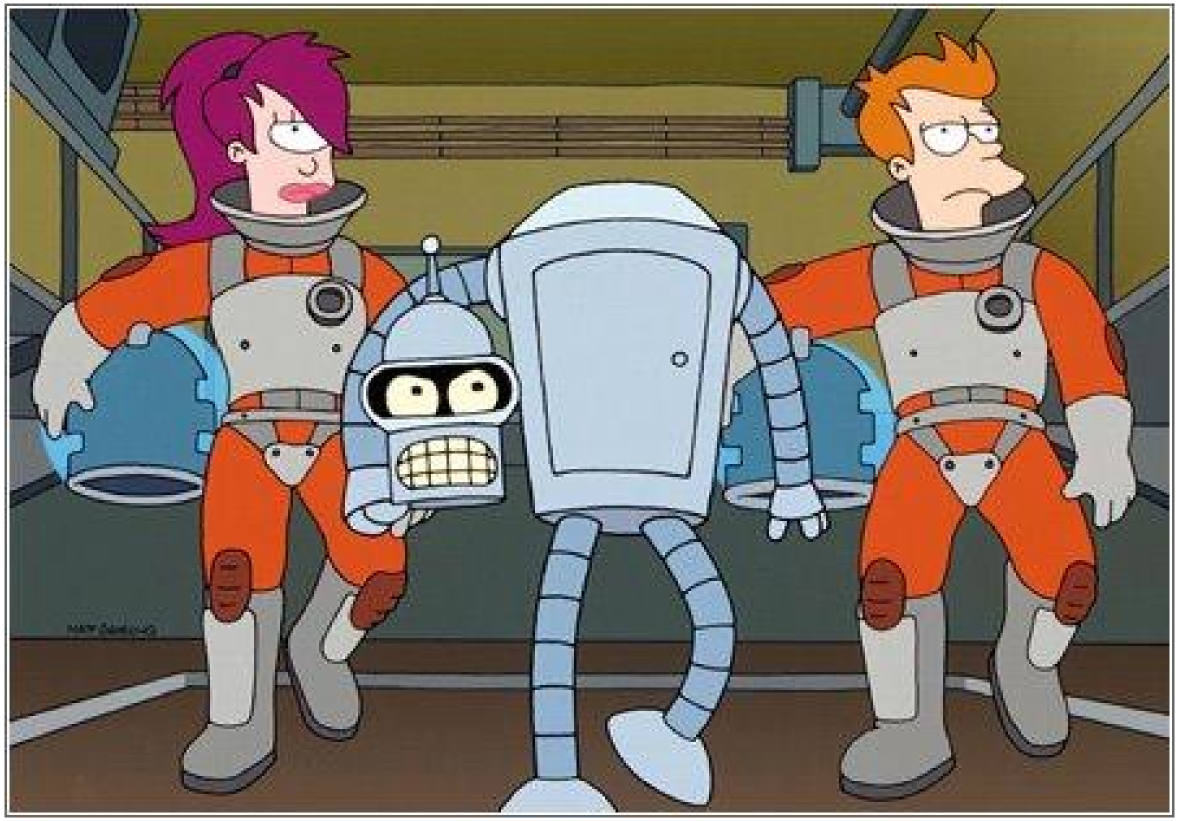
- Monitor for releases (Jenkins)
- Inspect the contents of released archives
- Catalog
  - Release Information
  - JAR metadata - versions/checksums/manifest
  - Class files
  - Fingerprints
  - Embedded dependencies

The How? - The Full Picture



The How? - The Full Picture

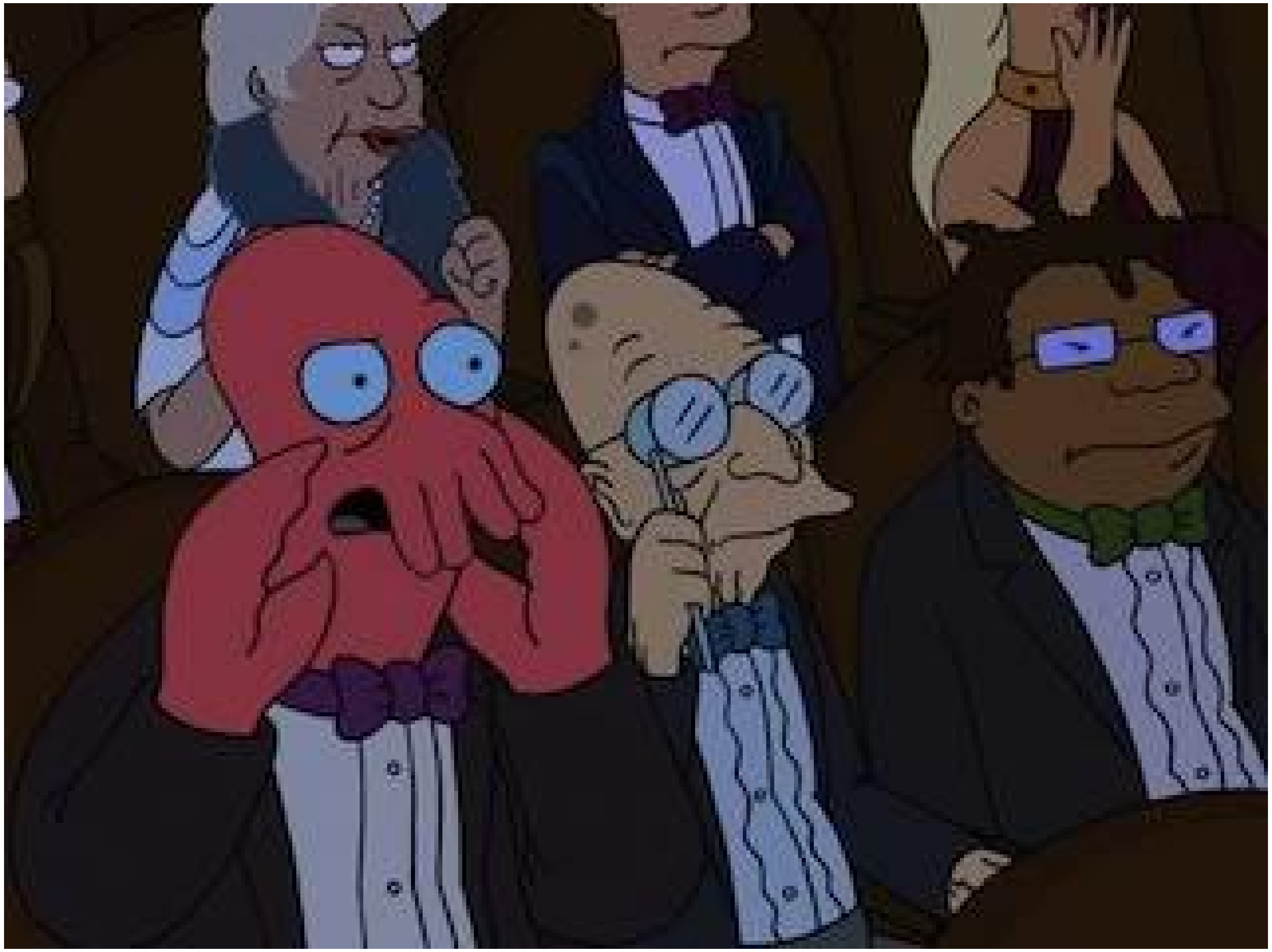
Live Preview Time



The Road Ahead

- Fine grained matching
  - JAR metadata can change from build to build
  - Determine similar classes?
    - One line fixes become problems
    - Compiler optimizations etc
- More contributions to the Victi.ms database
- Feedback from developers

Questions? Thoughts?



References

1. https://www.aspectsecurity.com/uploads/downloads/2012/03/Aspect-Security-The-Unfortunate-Reality-of-Insecure-Libraries.pdf
2. http://www.sonatype.com/Products/Why-Sonatype/Reduce-Security-Risk/Security-Brief
3. http://tvmedia.ign.com/tv/image/article/109/1095877/futurama-season-6-20100609103054764.jpg
4. http://twilight.ponychan.net/chan/noponycares/src/132953243859.jpg
5. http://qfxblog.files.wordpress.com/2009/08/futurama1.jpg