Openly Handling Security Vulnerabilities

Moderator and Panelist: Igal Sapir
Panelists: Christopher Schultz, Coty Sutherland, Mark Thomas

Apache Tomcat PMCs
Openly Handling Security Vulnerabilities

- Tomcat is an Apache project
- Apache Software Foundation mandates public communications (i.e. mailing lists)[1]
- Bug tracking[2], revision control[3] are public

Openly Handling Security Vulnerabilities

- Commits are immediately visible to the world
- Must commit before rolling a release
- Releases votes require 3 days to collect votes
- How do we securely fix security issues?
Openly Handling Security Vulnerabilities

• Community Reports
  – Bug tracker / GitHub :(
  – Twitter :(
  – BlackHat, DefCon, etc. :( 
  – Press :( 
  – Bug Bounty programs (e.g. EU-FOSSA) 
  – security@tomcat.apache.org :)
Openly Handling Security Vulnerabilities

- Private discussion (mailing list)
  - Is the vulnerability actually valid?
  - Determine severity
    - Who is/can be affected?
    - How bad could effects be?
    - Requires a CVE [1]?
    - Possible mitigations?

[1] “Common Vulnerabilities and Exposures”
Openly Handling Security Vulnerabilities

- Obtain CVE (if appropriate)
- Patch, Vote, Release
- Announce
Openly Handling Security Vulnerabilities

• Announcements
  – Mailing lists (users@, dev@, announce@)
    • Look for [SECURITY] in subject
  – Project page[1]
  – Will include mitigations
  – Will not include full disclosure, PoC, pen tests, etc.

Openly Handling Security Vulnerabilities

- Obfuscating patches
  - Sometimes difficult
  - Sometimes unnecessary
Examples

- Netflix 8x http/2 vulns
  - Privately contacted Tomcat project 2019-05-23
  - Most http/2 implementations affected (httpd, nginx, etc.)
  - Tomcat was somewhat susceptible to 1 of 8 vulns
    - CVE-2019-9513 HTTP/2 Resource Loop
Examples

• Netflix 8x http/2 vulns
  - CVEs were already assigned
  - “Easy” to exploit, basic DOS
  - ...which wasn’t any worse than making a typical http/2 request to Tomcat
  - Tomcat security team decided this wasn’t a vuln
Examples

• Netflix 8x http/2 vulns
  - Responsibly disclosed
  - Nicely coordinated with other vendors
  - Announced once patches had been available for all affected products
Examples

● Chaitin AJP Attribute-Injection
  – Privately contacted Tomcat project 2020-01-03
  – Some question as to whether or not this was a vuln
    • Attribute-injection is a feature of AJP
    • Publicly-exposed AJP is insanely insecure
Examples

- Chaitin AJP Attribute-Injection
  - Fixed in source repo 2020-02-04
  - Released 9.0.31, 8.5.51 2020-02-11
    7.0.100 2020-02-14
  - Announcement planned for 2020-03-14 (7.0.100 + 4wk)
  - CNVD announced 2020-02-20 :
  - Apache Tomcat Security Team forced to announce 2020-02-24
Responsible Disclosure

- Coordinated Disclosure
- Gives security team time to evaluate, mitigate
- Helps keep users safer
- Don’t worry, you’ll still get credit
  (And you can make up a catchy name, too, if you want)
Responsible Disclosure

- Contact security@tomcat.apache.org (or security@apache.org, we’ll get it)
- Clear explanation with PoC is best
- Remain engaged
- Respect any disclosure-embargo
  - Remember: not everyone can upgrade on release-day
Q&A with Attendees

Please ask your questions in the chat; the moderator will choose questions for the panel.
Sample Topics

• What counts as a security vuln versus just a bug?
• Does DOS count as a security vuln?
• Java doesn’t use pointers: aren’t most security vulns impossible?
• How do other OSS projects approach security?