What Is Bug Triage?

- *Bug Triage* is the process of reviewing open bug reports to make sure that they are:
  - reported in the correct place
    - Correct component
    - Something that Fedora can fix
  - in the correct status
  - detailed enough to aid the package maintainer in fixing the bug
  - not a duplicate of a previously reported bug
  - not already fixed

## Report: Version / Status

### As of 2008-06-20

<table>
<thead>
<tr>
<th>Version</th>
<th>ASSIGNED</th>
<th>FAILS_QA</th>
<th>MODIFIED</th>
<th>NEEDINFO</th>
<th>NEW</th>
<th>ON_DEV</th>
<th>ON_QA</th>
<th>POST</th>
<th>RELEASE_PENDING</th>
<th>VERIFIED</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>909</td>
<td>2</td>
<td>120</td>
<td>257</td>
<td>1813</td>
<td>3</td>
<td>62</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>3184</td>
</tr>
<tr>
<td>9</td>
<td>1459</td>
<td></td>
<td>216</td>
<td>200</td>
<td>2451</td>
<td>4</td>
<td>96</td>
<td>22</td>
<td>3</td>
<td>4</td>
<td>4455</td>
</tr>
<tr>
<td>rawhide</td>
<td>645</td>
<td></td>
<td>50</td>
<td>67</td>
<td>1513</td>
<td>1</td>
<td>14</td>
<td>2</td>
<td>.</td>
<td>.</td>
<td>2292</td>
</tr>
<tr>
<td>Total</td>
<td>3015</td>
<td>2</td>
<td>386</td>
<td>524</td>
<td>5779</td>
<td>8</td>
<td>172</td>
<td>34</td>
<td>7</td>
<td>8</td>
<td>9935</td>
</tr>
</tbody>
</table>
Anybody Can Help

You do NOT need to:

- understand individual bug reports and solve them yourself.
- be a programmer or package maintainer
- commit a significant number of hours each day or week to have an impact

You DO need to have a:

- basic familiarity with Fedora and Linux in general
- basic understanding of how RPM packages work
- desire to learn more and make Fedora better
Bug States

- The foundation of bug triage is built on the status of each bug and helping to make sure they reach their final resting place—CLOSED.

- Our current focus is bugs in states of:
  - NEW
  - NEEDINFO

- To be an effective triager it also helps to have an overall understanding of “the life of a bug”
  - See the next slide
Bug State Flow

This picture shows the normal states a bug goes through in Fedora.

Regular Triage Duties

- Locating and reviewing bugs with a status of:
  - NEW
  - NEEDINFO
- Requesting more information or changing the status of the bug to its next state
  - See previous flowchart to determine the next correct state
- Reviewing bugs in *unusual* states
  - States not usually used by Fedora
  - See diagram on previous slide
Triaging NEW Bugs

- Locate bugs with a status of NEW
- Perform a general review of the bug to make sure that:
  - It is reported against a supported version of Fedora
  - Contains enough information for the package maintainer to investigate the cause of the bug
  - Is reported against the correct component
  - Is not a duplicate of an existing bug
- If everything is correct, the bug should be changed to ASSIGNED
- If information is missing and needs to be requested, add a comment and change the bug to NEEDINFO
- If the bug is a duplicate or already fixed, add a comment and change the bug to CLOSED
Bugs that Fedora CANTFIX

- There are some bugs that Fedora can't fix
  - Software that we don't ship
  - Proprietary drivers not working
    - nVidia
    - VMWare
    - ATI fglrx (NOT radeonhd)
- Tainted or custom kernels (only for kernel bugs)
Finding Duplicate Bugs

- Use https://bugzilla.redhat.com/query.cgi?format=specific
  - Most effective method of finding dupes
  - Also the simplest search method :)  
  - Search open bugs first, then closed
- Judicious use of keywords is essential
  - Too broad and you have thousands of bugs
  - Too narrow and you won't find a potential duplicate
- If the problem refers to specific hardware, then searching on either the name of the hardware or the driver can be helpful
  - Not so helpful if driver is really common
Duplicate Bugs – continued

- There is a facility at http://bugz.fedoraproject.org/<package> where you can view a list of ALL open bugs against a package
  - Useful for components with a small number of bugs
- For packages with very low amounts of open bugs, you can use this interface and scan summaries to find dups
- However, this won't help you find incorrectly filed bugs (i.e. against the wrong component.)
Triaging NEEDINFO Bugs

- Locate bugs with a status of NEEDINFO
- If a bug has been in NEEDINFO for more than thirty (30) days and there has been no response to the requested information
  - Add a comment noting that there has been no response
  - Change status of the bug to CLOSED
- Standard wording and several polite ways of conveying this message are here:
- If the information has been provided, but the state has not been changed to the previous state, change the bug to the appropriate state.
You Want to Jump In?

Here is what you need:

1) Fedora Account
2) Add 'fedorabugs' group membership to your Fedora Account
3) Sign up for a Red Hat Bugzilla account
   - email account (buzilla user id) must match Fedora Account
4) Add your name to the Active Triager's wiki page
   - http://fedoraproject.org/wiki/BugZappers/ActiveTriagers

- All the current details are here:
Finding Bugs

- There's going to be a live demonstration of this
- RSS feeds
- Pre-formatted queries found at above link—or request one
- Columns displayed in bugs can be changed, I use Bug ID, creation date, change date, assignee, state, version, component, and short summary. This winds up looking like this:

<table>
<thead>
<tr>
<th>ID</th>
<th>Open Date</th>
<th>Changed Date</th>
<th>Assignee</th>
<th>Status</th>
<th>Comp</th>
<th>Vers</th>
<th>StatusSummary</th>
</tr>
</thead>
<tbody>
<tr>
<td>435871</td>
<td>2008-03-04</td>
<td>2008-05-01</td>
<td><a href="mailto:john@ncphotography.com">john@ncphotography.com</a></td>
<td>NEW</td>
<td>bugzilla 8</td>
<td>SELinux is preventing createaccount.c (httpd_bugzilla_scr...</td>
<td></td>
</tr>
<tr>
<td>436536</td>
<td>2008-03-07</td>
<td>2008-05-01</td>
<td><a href="mailto:john@ncphotography.com">john@ncphotography.com</a></td>
<td>NEW</td>
<td>bugzilla 8</td>
<td>can't connect to secure LDAP server for auth</td>
<td></td>
</tr>
<tr>
<td>437045</td>
<td>2008-03-11</td>
<td>2008-05-01</td>
<td><a href="mailto:john@ncphotography.com">john@ncphotography.com</a></td>
<td>NEW</td>
<td>bugzilla 8</td>
<td>LDAP users can log in even if their accounts have expired</td>
<td></td>
</tr>
</tbody>
</table>
Tools of the trade

- Don't be afraid, you needn't know about all of these, but you'll probably encounter them:
  - Bugzilla, the bug tracking database for Fedora
  - The package database, pkgdb
  - The updates system, bodhi
  - The buildsystem, koji
Triaging a bug

- Let's take the first bug in the last screen shot as an example, bug 435871. This is a bug about an SELinux denial.

- Looking at the bug, we note that it has an AVC message:

- At the top of the bug is the component that this is assigned to right now

- The component of this should be selinux-policy-mls
Component guidelines

- Components in bugzilla are SRPM names, not binary RPM's. For example, if a person is having a problem with nash, there is no component for that. Find the SRPM name via 'rpm -qif <filename>'. For nash, that's mkinitrd

- The component of a bug should be the thing that is actually responsible for the issue. The previous bug was originally (incorrectly) filed against bugzilla.

- The bugzilla package has nothing to do with SELinux policy that is enforced against it, therefore the component should be the SELinux policy in use

- We notice from the AVC message that the policy in use is the targetedMLS policy, therefore, the bug should be assigned to selinux-policy-mls. If you are unsure, assign SELinux bugs to either selinux-policy-targeted or just plain selinux-policy
Searching Bugzilla

• There's nothing to be afraid of :)
• The advanced search form looks like a jumble of things that some web designer did as their first project and then threw a bunch of JavaScript around it to make it somewhat prettier, but it really is very functional
• There are two other methods of searching (at current, one may be going away in Bugzilla 3.2 – the simple search)
  – Specific
  – Simple
• The specific search searches both subjects and comments for keywords. This is a simple, yet powerful search
• The simple search allows you to narrow by component
### Searching Bugzilla

#### Search for Bugs (Advanced)

[Specific] [Simple] [Advanced] [Duplicates]

Give me some help (reloads page.)

#### Quick Search

**Summary** contains all of the words/strings

**Search**

#### Product Information

<table>
<thead>
<tr>
<th>Products</th>
<th>Versions</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugzilla</td>
<td>All</td>
<td>All</td>
</tr>
<tr>
<td>Dogtag Certificate System</td>
<td>1.0</td>
<td>4Suite</td>
</tr>
<tr>
<td>eCOS</td>
<td>1.0-beta1.0-beta2</td>
<td>BKingdoms</td>
</tr>
<tr>
<td>Fedora</td>
<td>1.0.1</td>
<td>915resolution</td>
</tr>
<tr>
<td>Fedora Directory Server</td>
<td>1.0.2</td>
<td>9wm</td>
</tr>
<tr>
<td>Fedora Documentation</td>
<td></td>
<td>ABE</td>
</tr>
<tr>
<td>Fedora EPEL</td>
<td></td>
<td>AGReader</td>
</tr>
</tbody>
</table>

#### Status, Type, and Platform Information

<table>
<thead>
<tr>
<th>Status</th>
<th>Resolution</th>
<th>Severity</th>
<th>Priority</th>
<th>Platform</th>
<th>OS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW</td>
<td>NOTABUG</td>
<td>urgent</td>
<td>urgent</td>
<td>All</td>
<td>Linux</td>
</tr>
<tr>
<td>ASSIGNED</td>
<td>WONTFIX</td>
<td>high</td>
<td>high</td>
<td>i386</td>
<td>All</td>
</tr>
<tr>
<td>NEEDINFO</td>
<td>DEFERRED</td>
<td>medium</td>
<td>medium</td>
<td>alpha</td>
<td>All</td>
</tr>
<tr>
<td>MODIFIED</td>
<td>WORKSFORME</td>
<td>low</td>
<td>low</td>
<td>sparc</td>
<td>Windows</td>
</tr>
<tr>
<td>ON_DEV</td>
<td>CURRENTRELEASE</td>
<td></td>
<td></td>
<td>noarch</td>
<td>Mac OS</td>
</tr>
<tr>
<td>ON_QA</td>
<td>RAWHIDE</td>
<td></td>
<td></td>
<td>ia64</td>
<td>Other</td>
</tr>
<tr>
<td>VERIFIED</td>
<td>ERRATA</td>
<td></td>
<td></td>
<td>i586</td>
<td></td>
</tr>
<tr>
<td>FAILS_QA</td>
<td>DUPLICATE</td>
<td></td>
<td></td>
<td>i686</td>
<td></td>
</tr>
</tbody>
</table>

#### String Search

- **A Comment** contains the string
- **The URL** contains all of the words/strings
- **Whiteboard** contains all of the words/strings
- **Fixed In Version** contains all of the words/strings
- **OA Whiteboard** contains all of the words/strings
Advanced Search

- We don't use all of the fields!
  - Platform and OS are generally not used
  - Severity and Priority are ignored (for now)
- This reduces the apparent complexity of the form greatly
- Any field that has no selection is treated as though it didn't exist – you don't have to fill out every possible field
- There are some powerful limiters available to use – including full-text comment search.