Advancing the SIP Standards -Tracking-

Robert Sparks
Estacado Systems
SIPit

- Twice a year - moves between continents
- 100s of implementations
SIPit

- SIPit 16 statistics
  - 2 major specification bugs
  - A dozen specification clarifications
  - Hundreds of implementation errors
SIPit

• Opportunity to observe
  – What’s being implemented
  – What’s not being implemented
  – What folks are trying that’s not working
  – What’s being built without standardization

• Opportunity to listen to
  – What implementers find important
  – How the specs are interpreted
    • (when does MUST mean MUST, and why should we ever implement a SHOULD?)
What’s getting implemented?
What should we track?

• Send me email with what should be on the SIPit 18 survey
  – GRUU, outbound, connect-reuse
  – TLS, sips
  – Identity, sipping-certs
  – Non-trivial offer-answer
  – More details on use of PRACK
Bug Tracking

• Bugzilla - http://bugs.sipit.net
• Heavily utilized during 3261 edits
• Only draft editor can add/modify bugs
• Two different use models
  – Fixing bugs while editing drafts
  – Identifying bugs to fix in future spec updates

<table>
<thead>
<tr>
<th></th>
<th>3261</th>
<th>3262</th>
<th>3263</th>
<th>3264</th>
<th>3265</th>
<th>3420</th>
<th>3515</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Bugs</td>
<td>91</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>18</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Building interop reports

- **Huge** task - only tractable in increments
- From those docs in the hitchhiker’s guide:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2607</td>
<td>MUST</td>
</tr>
<tr>
<td>1386</td>
<td>SHOULD</td>
</tr>
<tr>
<td>1188</td>
<td>MAY</td>
</tr>
<tr>
<td>500</td>
<td>MUST NOT</td>
</tr>
<tr>
<td>250</td>
<td>RECOMMENDED</td>
</tr>
<tr>
<td>222</td>
<td>SHOULD NOT</td>
</tr>
<tr>
<td>82</td>
<td>OPTIONAL</td>
</tr>
<tr>
<td>72</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>37</td>
<td>NOT RECOMMENDED</td>
</tr>
</tbody>
</table>
Building interop reports

- Two approaches
  - Identify subset of each spec where it makes sense to pursue a report
  - Provide a place for the community to volunteer reports ad-hoc and watch for critical mass
New Tools?

- Different systems for spec issues vs. bugs in drafts being developed
- General wiki for tracking state of implementation and interop statements