Revisiting Unix principles for modern system automation

Martin Krafft
Just because the Cloud is all hip and stuff doesn't mean that you need to reinvent the wheel all the time thank you very much.
reclass inventory management

• Parametrise knowledge about your hosts
• Parameter merging up hierarchies (i.e. how LDAP/X.500 got it wrong)
• Supports Puppet, Ansible, Salt; adapters easy to write
• Currently working, but development stalled
• http://reclass.pantsfullofunix.net
SSH “botnet”

- Persistent connections between nodes (spoke, bouncers, mesh?)
- Run sysadmin stuff on the “botnet”
- Focus on transport.
  Do that well.
  Provide interfaces.
  Be Unix-y.
“OSI Stack“ of system automation

- Enactment
- Policy
- Transport
- Inventory (reclass)

Authentication
Encryption
Topology
Resilience
## Transport Feature Comparisons

<table>
<thead>
<tr>
<th></th>
<th>cfengine</th>
<th>Puppet</th>
<th>Chef</th>
<th>Salt</th>
<th>Ansible</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authentication</strong></td>
<td>NIH</td>
<td>NIH</td>
<td>NIH</td>
<td>NIH</td>
<td>SSH</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>NIH (TLS)</td>
<td>NIH (TLS)</td>
<td>NIH (TLS)</td>
<td>NIH (crap)</td>
<td>SSH</td>
</tr>
<tr>
<td><strong>Topology</strong></td>
<td>Pull</td>
<td>Pull</td>
<td>Pull</td>
<td>Pull or hope</td>
<td>Push</td>
</tr>
<tr>
<td><strong>Resilience</strong></td>
<td>Good</td>
<td>Fair</td>
<td>Fair</td>
<td>Hope-based</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Transport Unix style

- Do one thing and do it well
- Topology independent: push \textit{and} pull
- Plain interfaces, so usable by other tools that do e.g.
  - Monitoring
  - Data collection, asset management
  - Remote execution, policy enforcement
Live demo
ssh://git@github.com/madduck/retrans

Thanks!

Martin Krafft
<madduck@debian.org>