Centralised patch management
What did we do?

- Look at client update tools
- Create list of research topics
- Investigate three existing patch management systems
- Compose list of functional requirements for ideal patch management
- Build Proof of Concept
## Existing systems

<table>
<thead>
<tr>
<th>Area</th>
<th>WSUS</th>
<th>Radia</th>
<th>LANDesk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patches</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>End users</td>
<td>+</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Distribution</td>
<td>0</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Administration</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>User interface/Framework</td>
<td>++</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>0</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Reporting</td>
<td>0</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

### Contents

- Activities
- Comparison
- Requirements
- Proof of Concept
- Conclusion
Ideal requirements (1/5)

Patches:

- Acquire via existing mechanisms or a third party
- Rollback capability
- Verification (digital signature, checksum)
- Multi-platform
  - Impossible to support everything
  - Multiple PMS's is not a bad thing
Ideal requirements (2/5)

End users:

- Should not be able to reject or rollback patches
- Reboot options should be versatile:
  - Warning
  - Postpone
  - Deadline
  - After office hours
Ideal requirements (3/5)

Distribution:
- Agent & existing mechanisms
- Prioritization (based on risk / severity)
- Grouping of hosts (servers / workstations)
- “One, some, many”

Administration:
- Approve / reject patches
- Custom patches / scripts
Ideal requirements (4/5)

User interface / Framework:

- User-friendliness
- Access control
- Backups / restore
- More information about patches (CVE)

Infrastructure:

- Multicast / peer-to-peer / multiple servers
- Low / expensive bandwidth users
- Inventory building
Ideal requirements (5/5)

Reporting:

- Alerting (SMS, e-mail, etc)
- Reports
  - Patches (success, failure, new, rejected, etc)
  - Hosts (completely patched, missing patches)
  - Groups (hosts, approved patches)
  - . . .
Proof of Concept

- Why a Proof of Concept?
- Why APT?
- Why Ubuntu?
Contents
➢ Activities
➢ Comparison
➢ Requirements
➢ Proof of Concept
➢ Conclusion
APT module

• Synchronize
  ➔ Download Package & Release file
  ➔ Verify signature & checksums
  ➔ Store package info in database

• Build
  ➔ Retrieve package info from database
  ➔ Make new Package & Release file
  ➔ Create digital checksums & signature
Conclusion

- Product investigation
- Ideal requirements
- Proof of Concept
- Future work
Questions?
Screenshots (1/3)
Screenshots (2/3)

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2</td>
<td>1.0.2-2ubuntu0.1</td>
</tr>
<tr>
<td>cvs</td>
<td>1:1.12.9-9ubuntu0.1</td>
</tr>
<tr>
<td>gdb</td>
<td>6.3-5ubuntu1.1</td>
</tr>
<tr>
<td>gzip</td>
<td>1.3.5-9ubuntu3.2</td>
</tr>
<tr>
<td>sudo</td>
<td>1.6.8p5-1ubuntu2.1</td>
</tr>
</tbody>
</table>

Build repository
Screenshots (3/3)

```
root@uilskuiken:~ # apt-get update
Get:1 http://145.92.27.32 hoary-security Release.gpg [189B]
Hit http://145.92.27.32 hoary-security Release
Hit http://145.92.27.32 hoary-security/main Packages
Fetched 1B in 0s (29B/s)
Reading package lists... Done
root@uilskuiken:~ # apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
The following packages will be upgraded:
  gzip
1 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/70.3kB of archives.
After unpacking 0B of additional disk space will be used.
Do you want to continue [Y/n]? 
```