PANIC/PYALARM AT SOLARIS – S2INNOVATION IN ACTION

Piotr Goryl, Łukasz Żytniak, S2Innovation Sp. z o.o., piotr.goryl@s2innovation.com

2018-06-07, Tango Collaboration Meeting, ELI Beamlines
Outline

- Scope
- Workflow
- Deployment
- Development
- Remarks
Scope

- PANIC/PyAlarm deployment
  - Mails
  - SMS
  - SNAP
- Gather requirements and configure 140 alarms
- Integration with eLog
- SVG Synoptic update with alarm layer
- Web application
- Training
Workflow

- Initial presentation/training
- Requirements gathering - done
  - Create a spreadsheet
  - Interviews and discussion on philosophy -> limit number of alarms, do not duplicate interlock application
  - Get meaningful description of alarm situation detection
- Prepare formulas - done
- Use a custom scripts to configure the system - done
- Perform tests – in progress
  - Activate alarms
  - Update formulas and levels if needed
  - Tests are in progress
• First approach was to use Conda environments to avoid conflicts with existing packages (numpy, fandango).
• In meantime SOARIS migrates control room to use Ansible and rpms based distribution, so we adopted and PANIC is now installed from an rpm.
• On server side the PyAlarm rpm has been built, too.
• 19 PyAlarm devices are distributed over 12 virtual machines.
• Versions used:
  • GUI – PANIC 6.5.0-solaris -> to be updated to 6.5.1
  • PyAlarm – PANIC 6.5.0.1-solaris (running script with PYTHONPATH unset )
    • Taurus 4.3.1 – from PyPi
    • PyTangoArchiving 8.1.3 – from GitHub
tango-controls/PANIC

development branch

master branch (tag)

S2Innovation/PANIC

SOLARIS feature or fix branch

development branch

SOLARIS rpm branch

https://github.com/S2Innovation
- Plugin djangoCMS PanicActiveAlarms - displaying table of active alarms
- Plugin djangoCMS PanicAlarmsHistory - displaying table of alarms' history
- Plugin djangoCMS PanicAlarmsFilters - allowing to filter the tables of alarms
ELogSender device class

- DevVarString argin
- que size is limited to prevent flooding

• Entry build is configured by properties

**PyAlarm device**

**PyAlarm device**

**PyAlarm device**

**ACTION(...)**

**creat_entry**

**Entries que**

**send_entry**

**elog command call**

**que size is limited to prevent flooding**

**Device properties [alarm/unit/elogsend1]**

<table>
<thead>
<tr>
<th>Property name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntryAttributes</td>
<td>Subsystem={%3%}</td>
</tr>
<tr>
<td></td>
<td>Level={%4%}</td>
</tr>
<tr>
<td></td>
<td>Stage={%5%}</td>
</tr>
<tr>
<td></td>
<td>Subject={%2%}</td>
</tr>
<tr>
<td>EntryMessage</td>
<td>%Queue Message%</td>
</tr>
<tr>
<td>EntryTitle</td>
<td>%LogbookName%</td>
</tr>
<tr>
<td>LogbookName</td>
<td>%0%</td>
</tr>
</tbody>
</table>
• Extension to Solaris application based on MAX-IV svg synoptic
  https://github.com/MaxIV-KitsControls/lib-maxiv-svgsynoptic
• Support for boolean attributes to present state
• Open custom window when clicked (AlarmForm)
Updates to PANIC

- Fixing new AlarmForm window - done
- Enabling ‘-r’ option for mail – done (not yet pull requested)
- Wiki link in AlarmForm – done (not yet pull requested)
- Documentation updates – planed
  - Mail, SMS – how-to
  - Recipes
Some remarks

• Very good support
• Few times we were looking to code to find information on how to configure certain functionalities (sms, mail)
• We have tried to run PANIC with Python 3.6, succeeding to run PyAlarm device server (without Taurus)
  • futurize of PANIC, PyTangoArchiving 7.8 and fandango
  • Needed to change to MySQL-python to mysqlclient-python
  • Some manual changed to libraries import had to be applied
Thank you

Special thanks to:

- SOLARIS
- ALBA
- MAX-IV