Zero-effort Monitoring Support

University of Amsterdam
Network and System Engineering

Julien Nyczak

Supervisor: Rick van Rein, ARPA2.net
Introduction

• Linux shipped with large amount of packages
• systemd, the new init system
• Process information available through systemd
• SNMP, standardized monitoring protocol
Related Work

• Existing process monitoring solutions:
  ▫ Linux Process Monitoring with Nagios:
    • Not using SNMP
    • Plugin required on the host side
  ▫ SNMP plugin for Nagios `check_snmp_process.pl`
    • Uses Host Resources MIB (RFC2790)
    • MIB covers only running processes
    • MIB not aware of invalid process state
    • No need of subagent
Related Work (2)

• Existing process monitoring solutions:
  ▫ **UCD-SNMP-MIB**
    • Covers running processes and their state (running or not)
    • Specific `snmpd.conf` configuration on monitored host
    • No need of subagent
Research Questions

• How feasible it is to integrate service monitoring in a generic manner for different systems (e.g. Red Hat and Debian)?

• How can SNMP be used to relay service status to a monitoring station and be aware of changes to adapt to them in an automated way?
Background - systemd

- Developed in 2010 by Lennart Poettering
- New init system
- Uses unit files instead of old init shell scripts
- Unit status can be queried with `systemctl` command
Background - The AgentX Protocol

- Standard for master and subagent communication
- Subagent not aware of SNMP traffic
- Has access to management information
- Registers OIDs with the master agent
- Binds OIDs with variables
Requirements for Automatic Service Monitoring

• Linux packages with a unit file

• Subagent built upon NET-SNMP -> tool packaged in rpm, with NET-SNMP as a dependency

• Started by default by systemd at boot time
Proof of Concept - Subagent

- Written using the python-netsnmpagent Python module developed by Pieter Hollants licensed under GPL v3
- Written for the Network Service Monitoring MIB (RFC2788)
- 3 OIDs used under the applTable:
  - applIndex
  - applName
  - applOperStatus
Proof of Concept - Subagent (2)

- Queries ALL service units with systemctl commands
  - unit is active, active and enabled or inactive and should be: applOperStatus = 1
  - unit is active but not enabled but should be: applOperStatus = 3
  - unit in unknown status: applOperStatus = 3
  - if other state (inactive, failed): applOperStatus = 2

- Can be configured with files to fine-tune monitoring:
  - units NOT to be monitored
  - units to be started at boot time
  - units that must be down
Proof of Concept - Monitoring

- Nagios

- No existing perfect SNMP plugin
  - Modified version of `check_snmp_table.pl` by William Leibzon licensed under GPL v2
  - Called in a home-made shell script

- But “proper” plugin needed
Proof of Concept - Workflow

Monitored Host

- `not_monitored_units`
- `enabled_units`
- `units_to_be_down`

- `systemd_subagent.py` + `NETWORK-SERVICE-MIB`
  - Queries `systemd` + Binds MIB OIDs with Variables
  - Initiates AgentX Session + Registers MIB OIDs
  - Accepts AgentX sessions + Accepts OID Registrations

- `Snmpd Master Agent`
  - SNMP Traffic

- `Nagios Monitoring Station`
  - Nagios run with `start_nagios.sh`
Conclusion

- Zero-effort monitoring support idea possible
- systemd is generic enough
- All packages should have a unit file
- Tool could be packaged and started by systemd at boot time
- Network Service Monitoring MIB lacks status specific to systemd
Future Work

- Develop the subagent in C
- Create a MIB meant for systemd unit monitoring (status specific to systemd)
Thank you for your attention!

Questions?

Julien.Nyczak@os3.nl