perfSONAR Plugin Development

Kristofer Hallin, Magnus Bergroth

What?

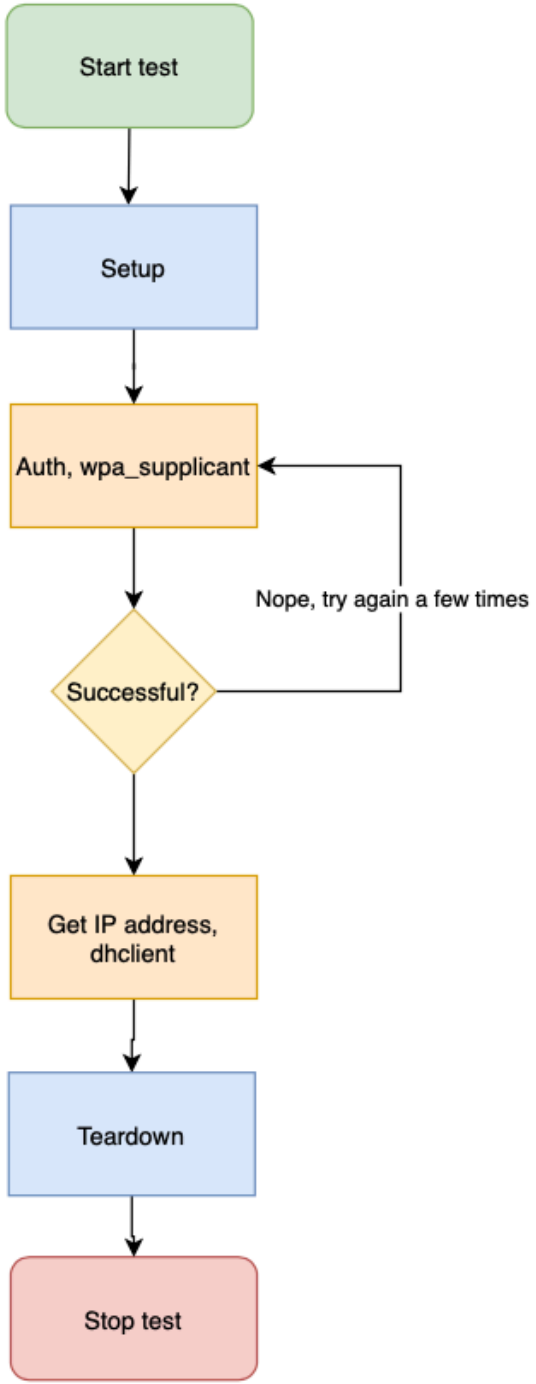
- We wanted a probe, easy to deploy and manage
- Standard Debian (Raspbian) on Raspberry Pi
- Easy installation, no manual intervention
- Easy to upgrade if needed

perfSONAR

SUNET

In its simplest form:

- Set up?
- Try to authenticate
- Try to get an adress
- Teardown?
- Run some other tests



SUNET

Auth

- Figure out how to configure things
- WPA Supplicant?
  - D-Bus directly? Nah.
  - wpa_cli?
- Ended up using a wpa_cli

```
root@raspberrypi:/home/pi# wpa_cli
wpa_cli v2.8-devel
Copyright (c) 2004-2019, Jouni Malinen <j@w1.fi> and contributors

This software may be distributed under the terms of the BSD license.
See README for more details.


Selected interface 'eth1'

Interactive mode

> reconfigure
OK
> add_network 0
0
> interface eth1
Connected to interface 'eth1.
> set_network 0 key_mgmt IEEE8021X
OK
> set_network 0 eap PEAP
OK
> set_network 0 identity "test"
OK
> set_network 0 password "test"
OK
> enable_network 0
OK
> reauthenticate
OK
<3>Associated with 01:80:c2:00:00:03
```

SUNET

DHCP

- Use ‘dhclient’ to get an address?
- How to prevent default route to be replaced?
- Network Namespaces to the rescue!
- Information about which address we got

SUNET

How to make a test and tool from this?

- There is a plugin development guide!

SUNET

Test

- Command line arguments, to and from specification
- Validation
- Result formatting (pretty-print)
- And more

```
pscheduler task --context '{"schema": 1, "contexts": [[{"context": "linuxnns", "data": {"namespace": "perfsonar"}}]]}' --tool wpa-supplicant 802.1x --interface eth1 --eap_type TTLS --username test --password test
```

SUNET

Tool

- Based on the information from the test, do the test
- Send back information (JSON) about the result
- Glue everything together in ‘run’
- Pass a test specification to it, over and over again

```
echo '{"test": {"spec": {"schema": 1, "wifi": {}, "timeout": "PT120S", "interface": "eth1", "_username": "test", "_password": "test", "eap_type": "TTLS"}}}' | ./run
```

SUNET

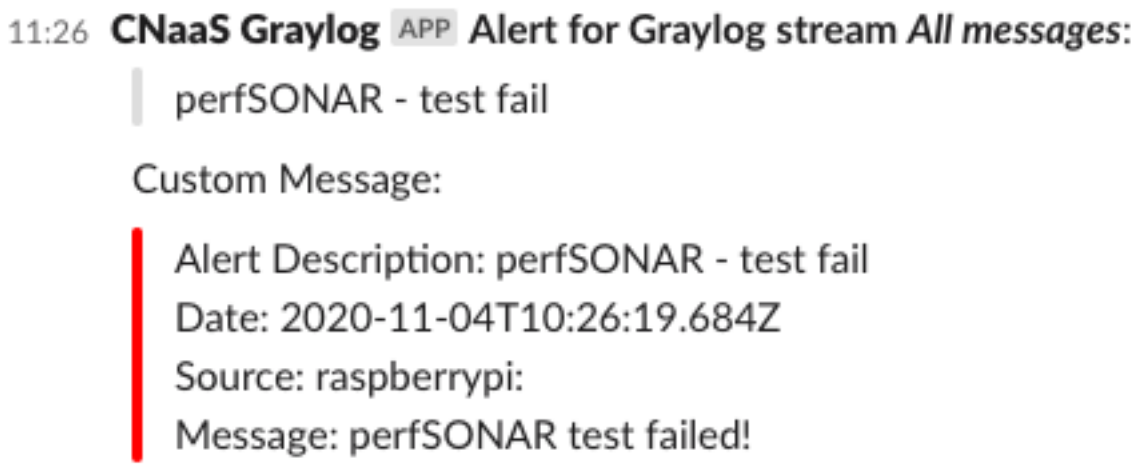
Result

```
pi@raspberrypi:~ $ pscheduler task --context '{"schema": 1, "contexts": [[{"context": "linuxnns", "data":
{"namespace": "perfsonar"}}]]}' --tool wpa-supplicant 802.1x --interface eth1 --eap_type TTLS --username
test --password test
Submitting task...
Task URL:
https://localhost/pscheduler/tasks/bce6a4d3-0ac3-48d2-bd84-0d19bd8739bd
Running with tool 'wpa-supplicant'
Fetching first run...

Next scheduled run:
https://localhost/pscheduler/tasks/bce6a4d3-0ac3-48d2-bd84-0d19bd8739bd/runs/a9e80eb7-
d60d-417d-8a84-9e4db0a5f135
Starts 2021-04-13T13:20:00+01 (~0 seconds)
Ends   2021-04-13T13:20:23+01 (~22 seconds)
Waiting for result...

BSSID ..................... 01:80:c2:00:00:03
MAC address ............... c4:41:1e:b4:c4:9f
IP address ................ 10.42.0.33

No further runs scheduled.
```

SUNET

Alarms, reporting etc

- Use the syslog archiver to send any results to Graylog
- No need to develop an archiver
- Graylog will send Slack notifications if the test fails



SUNET

Packaging

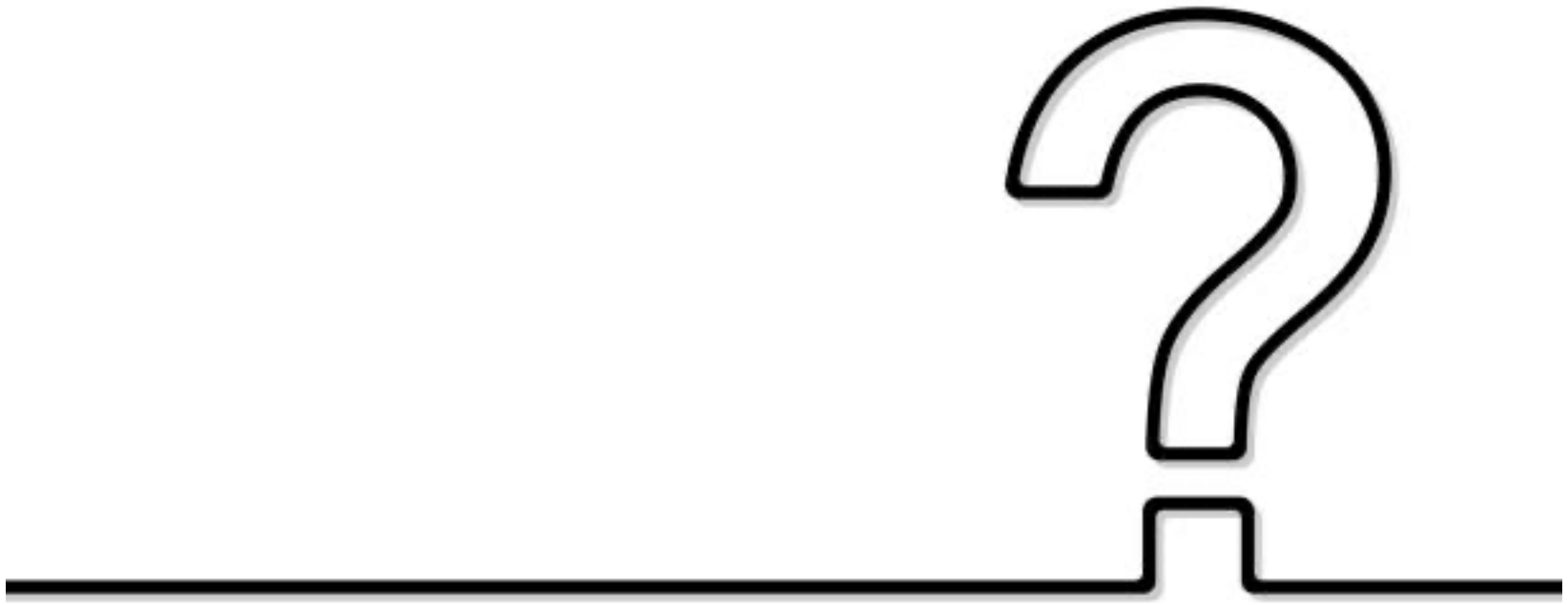
- Two Debian packages, test and tool
- Dependencies specified for each package
- The packages are responsible for all configuration
  - Create network namespace on boot
  - Attach network interface to it
  - Start ‘wpa_supplicant’ inside that namespace
- Must be possible to upgrade easily
- If merged upstream, this should be done in a better way

SUNET

Future

- Extend with support for WiFi stuff (channel, SSID etc)
- More generic package, let the tool take control of things
- Merge into pscheduler
- …

SUNET



SUNET