Exploring Solaris Auto-Registration

An irreverant look at evaluating new Solaris management tools

Tom Kranz

Some background first

- To manage infrastructure:
  - You need to know what all the bits are
  - You need to know what they do
  - You need to know why they do it
  - You need to know how they do it
  - Then you can start tuning/scaling/hacking

  New features/tools/applications/traps upset all of this

Here there be dragons

- So, an apparently new tool in the OS comes along
- 'secure' Internet upload of data?
  - From expensive bits of kit tucked away behind firewalls?
- I think not, chaps
- New bit of infrastructure, doing new things in new ways?
- Time for a poke around!

Why, Larry, why?

- Oracle need to make money from Solaris
- Oracle need to enforce their licensing
- Oracle need to know how and where Solaris is being deployed
- Oracle want to know what other products you're using
- Larry broke the mast on his yacht :-)

Fair's fair, though

- As a windsurfer, I'm totally with Larry wanting more cash from clients
  - Masts and sails are pricey (more so for yachts than windsurfers)
  - I'd totally gouge my clients for more windsurfing kit and time sailing
- As a sysadmin, I'm less impressed
  - What next? Clippy the Paperclip?
    - "Hi, I see you're deploying Oracle Solaris!"

So why mess with it?

- To know what's going on in my infrastructure
  - Is it secure?
  - Is it sensible?
  - Will it break something?
- Also, auto-reg broke my Jumpstart setup
  - Having it enabled by default irritated me
  - So I got hacking about to find out more

How does it work?

- The release notes are pretty good here
- It collects 'service tags' and uploads them to My Oracle Support
  - More on Service Tags at http://wikis.sun.com/display/ServiceTag/Sun+Service+Tag+FAQ
- Full list of data in a Service Tag is at:
  - https://inventory.sun.com/inventory/data.jsp

“We fear change”

- Actually, this existed before in Sun Inventory:
  - https://inventory.sun.com/inventory/
- And Service Tags plugged into Ops Centre
- And no-one really used it, because Explorer was all we cared about for support

“In the grim future, there is only OEM”

- OEM will consume all!
- Sun Ops Center has been absored into Oracle Enterprise Manager
- OEM doesn't just manage databases anymore
  - OS patch levels
  - Application deployments
- Like The One Ring, OEM Ops Center brings them all together and binds them

Simplify infrastructure management

- Everything gets linked in together with a coherent management platform
- CTOs love this stuff
- Beancounters don't – it costs a lot up front
- **But** you get the OEM bits by default when deploying Oracle databases
- This is the antithesis of system administration to a scruffy hacker like me

Argh! Make it stop

- OK, how to turn it all off?
- In Jumpstart:
  - Add autoreg=disable to sysidcfg
  - JET 4.8 has new template variables – key one:
    - base_config_sysidcfg_auto_reg=disable
- Interactive installs:
  - Get to da choppa^Wterminal!
    - Regadm disable
    - Or kill the SMF service svc:/application/autoreg:default

What about Solaris 11?

- Check out the 'Register Oracle Solaris' icon on the desktop
  - It calls /usr/bin/os-register
  - Which is a python script which talks to inventory.sun.com
  - It uses stclient, which is the CLI for service tag management

This all poses some issues

- I'm not really in the habit of deploying Solaris boxes in a corporate data centre with direct Internet access
  - Or via a proxy for that matter
  - And not if they're running RAC or similar critical loads
- SunInventory has a laptop client
  - Nasty cludge
  - I suspect it would make IDSs very unhappy too

Stclient

- Back in the days of Sun One, doing test installs of (eg.) Directory Server were problematical
- If you deleted it and tried to re-install it, you couldn't
- It used some sort of Java registry, and you had to delete the keys to re-install
- Egads! **stclient!**

Et tu, OpenIndiana?

```
bash-4.0$ uname -a
SunOS grond 5.11 oi_147 i86pc i386 i86pc
bash-4.0$ which stclient
/usr/bin/stclient
```

- /usr/bin/stclient -x dumps 4 service tags
- Yes, Alasdair is Mad Larry's stooge ;-)

Wait, it gets worse?

- Don't think that 'registering' will turn this off
- The SMF service stays enabled after registration
- After each boot, it scans for new service tags
- Then tries to upload them again

Let's hack about with it

- Stclient can remove service tags, so you can install something and delete the 'evidence'
  - This assumes the 'something' is not clever enough to respond to a subnet scan from another Solaris host
- We can also use stclient to make up **totally bogus** products that have been installed

The America's Cup is mine!

```
bash-3.00# stclient -a -p "Mad Larry's Yacht" -e
"2.0 + mast patch" -t 30b26c7d-15eb-4d81-
f546-dacc66b3aba3 -P Oracle -m Oracle -A
trimaran -z The_Sea -S A_Shipyard


Mad Larry's Yacht 2.0 + mast patch added
Product instance URN=urn:st:8986657f-b561-
c918-fafb-fa3de59e82c6
```

Now let's break HTTPS

- You'll be wanting ParosProxy for this
  - Nifty little Java proxy from www.parosproxy.org
- Extract it and run with java -jar paros.jar
- Configure regadm to use it:
  - Regadm set -n http_proxy -v localhost
  - Regadm set -n http_proxy_port -v 8080
- Then kick off a registration request
  - Regadm auth -u leo.apotheker@hp.com

Abbreviated message body

POST https://inv-cs.sun.com/SCRK/ClientRegistrationV1_1_0 HTTP/1.1

Content-Disposition: form-data; name="VERSION" 1.1.1

Content-Disposition: form-data; name="SOA_ID" leo.apotheker@hp.com

Content-Disposition: form-data; name="SOA_PW" password

Content-Disposition: form-data; name="ASSET_ID" 341214851

<and a public key attached here as well>

And the response?

```
TYPE=ERROR
CODE=4
MESSAGE=Cannot authenticate:
leo.apotheker@hp.com
 --
com.sun.scn.cs.usermgmt.client.NotFoundException: Not Found exception; method=POST;
key=session/leo.apotheker@hp.com?
source=SCRK; return code=404
```

Is it really that bad?

- You need to be root to mess with regadm/stclient
- The whole setup seems open to MITM attacks
  - Denial of service against a competitor? “Death by Oracle licensing?”
- Will the service tag scanning set off IDSs?
- Inventory management means licensing revenue – customers want some support advantage to this stuff too

Are these the end times that were foretold?

- It's clear the future of Solaris involves
  - Stricter licensing
  - Tighter integration into Oracle's software stack
- And this means more integration into management tools like OEM Ops Center
- Still bummed nothing seems to be leveraging Explorer though

Questions?

Or you can applaud, or throw coins, or something