



Michael DeHaan & Adrian Likins
Red Hat

What is Func?

A way to communicate with all your machines and query or control them over the network.

Func uses XMLRPC.

It's designed to be a way to write systems management apps for a very large number of systems.

Feature Overview

Remote API for systems management

Secure XMLRPC over SSL connections

Python API and CLI interfaces

(Python is great for system administration apps!)

SSL cert management tools, via 'certmaster'

But Wait, There's More!

Flexible machine name globbing, and user defined groups

Automates SSL certificate generation and deployment

- Can notify administrators of new requests

- Supports autosigning

- Integrates easily with kickstart & Cobbler

Modular, very easy to extend

Why Not Just Use SSH?

It's an API

- With return codes

- And error messages

- Structured data returned

You don't have to “\“\('figure out weird shell quoting')\”

Supported Platforms

Red Hat Enterprise Linux 4 and later (EPEL)

Fedora 6 and later

Debian*

FreeBSD*

(* Interested in getting them in the distro? Talk to us)

What Can You Do With It?

Use apps like FuncInventory and the Exploding Laptop Finder

Run arbitrary commands

Track files

Hardware profiling

Process info and control

Reboot and restart

Package install/update

Manage virtualized machines

Whatever else you like (user extensible w/ modules, can also call shell functions)

Func Architecture

Overlord

Minion

Certmaster

Func Minion

This is “funcd”

SSL XML-RPC daemon running on “client” machine (the machine you are controlling)

Local modules on each minion are surfaced over XMLRPC

Func Minion Modules

There is a Python interface for creating modules

Modules are simple, lots of examples to copy from

Some introspection available

- List the modules available remotely

- List the methods

- (pending) Get the method signatures and build UI programmatically

Func Overlord

This is the machine issuing commands

Can be multiple machines (just share the certs)

CLI script interface for now

Certmaster

Automates creation of SSL certificates

Can autosign if desired

Triggers are available to take action on receiving requests

Certificate created can be used by other apps

Usable without Func -- fedorahosted.org/certmaster

FuncInventory

An example of the power of the Func API:

```
# func-inventory [--help]
```

Ask every registered machine what modules it has

Asks what modules support inventory data

Saves inventory data per system using git

You get RSS feeds of the diffs

Only ~200 lines of Python!

Func command line

Examples:

```
func "foo*.example.org" call command run  
"/sbin/halt -p"
```

```
func "foo*.example.org" copyfile --file=/tmp/a  
--remotepath=/tmp/b
```

Custom Command Line Modules

Module API for creating custom CLI sub commands

No recompilation or rebuilds needed.

See “copyfile” for an example

API Demos

You've got a weird task from your boss.

You need to solve this really quickly.

Two arbitrary examples:

(A) generate a report of all the laptops you have that might explode

(B) generate a report of all the hard drives you have that are about to fail

Complete “Exploding Battery Finder” API Demo

```
import func.overlord.client as fc
import func.utils as utils
```

```
bad = open("./part_data.txt").read().split()
info = fc.Overlord("*").hardware.hal_info()
```

```
for (host,details) in info.iteritems():
```

```
    if utils.is_error(details):
```

```
        print "%s had an error : %s" % (host,details[1:3])
```

```
        break
```

```
for (device, full_output) in details.iteritems():
```

```
    for bad_value in bad:
```

```
        if full_output.find(bad_value) != -1:
```

```
            print "%s has flagged part: %s, matched %s" % (host, device, bad_value)
```

```
            break
```

Raw Data for “Exploding Battery Finder”

Just simple text:

92P1072 92P1073 92P1088 92P1089 92P1142 92P1141
92P1170", "92P1169",
93P5028 92P1174 92P1173 93P5030
1K055 C5446 F2100 KD494 W5915 Y1333 3K590 C6269
F5132 OR33X5308 Y4500 5P474
C6270 GD785 M3006 X5329 Y5466 6P922 D2961 H3191
RD857 X5332 C2603 D5555 J1524 TD349
X5333 C5339 D6024 JD616 U5867 X5875 C5340 D6025
JD617 U5882 X5877

Complete “Failing Hard Drive” API Demo

```
import func.overlord.client as fc
import func.utils as utils

info = fc.Overlord("*").smart.info()
failures = 0

for (host,details) in info.iteritems():
    if utils.is_error(details):
        print "%s had an error : %s" % (host,details[1:3])
        break
    (rc, list_of_output) = details
    if rc != 0:
        print "=====
        print "Host %s may have problems" % host
        print "\n".join(list_of_output[3:])
        failures = failures + 1

print "\n%s systems reported problems" % failures
```

Authorization and ACLS

Maps certificate ids to allowed methods

On certain servers, only allow certain overlords to run certain modules

The Future

Improved SSL features

- Cert revocation

GSOC student projects (thanks Google!)

- Extend funcweb, new modules, API enhancements

More modules

XMLRPC to Overlord

Get more apps using Func for network communication

World domination

Additional resources

Your comments, questions, bug reports, patches, and ideas are always welcome and help make Func what it is. Please share them with us.

hosted.fedoraprojects.org/func

#func, on freenode irc network

func-list@redhat.com

Q&A

...