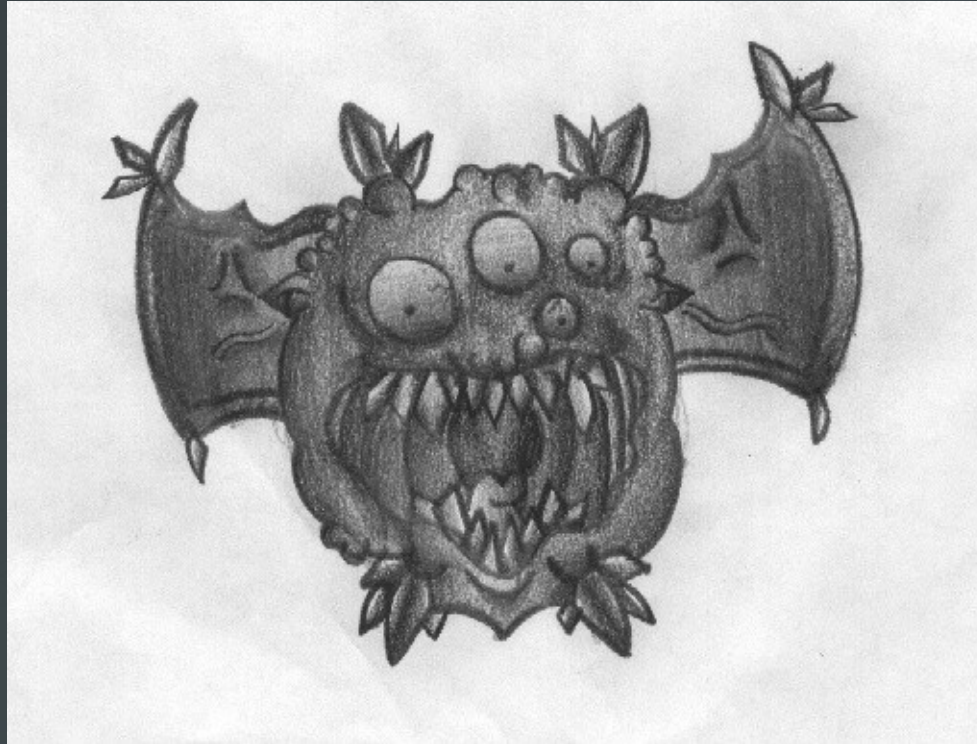


Writing UNIX Daemons with Working::Daemon



me?

Erwan Lemonnier

- perl since 2001
- Nordic Perl Workshops in Stockholm
- ERWAN on cpan

Glue Finance AB

- Hiring perl freelancers!! (talk to me :)



What's a daemon?

Wikipedia:

In Unix [...] a daemon is a computer program that runs in the background

Typically daemons have names that end with the letter "d"

That's pretty much it...



What do daemons do?

- Fork itself to background
- Log to system log
- Run as a less privileged user
- Run in a jail
- Start during boot
- Save its pid to file



Working::Daemon

- One of many CPAN daemon modules
- Arthur Bergman
- Still being developed
 - (logging does not work)
 - (the source is the pod)



daemon.pl

```
use Working::Daemon;  
our $VERSION = 0.45;  
  
my $daemon = Working::Daemon->new( );  
$daemon->name( "testdaemon" );  
$daemon->standard( );  
  
while (1) { sleep 60; }
```



Default arguments

```
[erwan@lookfar ~]$ ./daemon.pl --help  
[start | stop | restart | status]  
  --version  
  --group=str  
  --pidfile=str  
  --user=str  
  --no-daemon  
  --help  
  --foreground  
  --name=str  
  --loglevel=int  
  --no-chroot
```



start/stop/status

```
[erwan@lookfar ~]$ ./daemon.pl
info - Starting 'testdaemon'
testdaemon started on 49835
[erwan@lookfar ~]$
```

```
[erwan@lookfar ~]$ ./daemon.pl status
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
testdaemon is running on 49835
```

```
[erwan@lookfar ~]$ ./daemon.pl stop
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
49835 s003 S 0:00.04 testdaemon - waiting for child 49836
info - sent SIGINT to 49835 - waiting on stopped pid 49835
info - exiting master session testdaemon - child is 49836
info - Stopped testdaemon on 49835
```



Starting during boot

A script using `Working::Daemon` is already an `/etc/init.d` start/stop script

```
# ln -s /etc/init.d/daemon /your/bin/daemon.pl
```



Testing from terminal

```
$ ./daemon.pl --no-chroot --foreground --pidfile=./pid
```



Under the hood

```
$daemon->standard();
```

Really does:

```
$daemon->parse_options(@_);  
$daemon->do_action();  
$daemon->change_root();  
$daemon->drop_privs();
```



The undocumented API :)

Attributes:

user, group, pidfile, log_level, chroot, foreground,
daemon, name

Methods:

- `$d->parse_options()`
- `$d->change_root()`
- `$d->drop_privs()`
- `$d->parse_options()`
- `$d->write_pidfile()`, `$d->delete_pidfile()`
- `$d->spawn_worker_child()`
- `$d->daemonize()`
- `$d->is_running`
- `$d->get_pid`



Dropping privileges versus tainted mode

Dropping privileges:

```
$< = $> = $uid;  
$( = $) = $gid;
```

Or: `$daemon->drop_privs();`

Perl turns on tainted mode automatically.
System calls (open, unlink) will fail.

Solution: open filehandles before dropping privileges.



Dropping privileges versus tainted mode

```
use Working::Daemon;
our $VERSION = 0.45;

my $daemon = Working::Daemon->new();
$daemon->name('testdaemon');
$daemon->user('testdaemon');
$daemon->group('nobody');
$daemon->pidfile("/tmp/testdaemon.pid");
$daemon->chroot(0);

$daemon->parse_options(@ARGV);
$daemon->do_action();

# open files, set signal handlers...

# now ok to switch to tainted mode:
$daemon->drop_privs();

while (1) { sleep 60; }
```



What next?

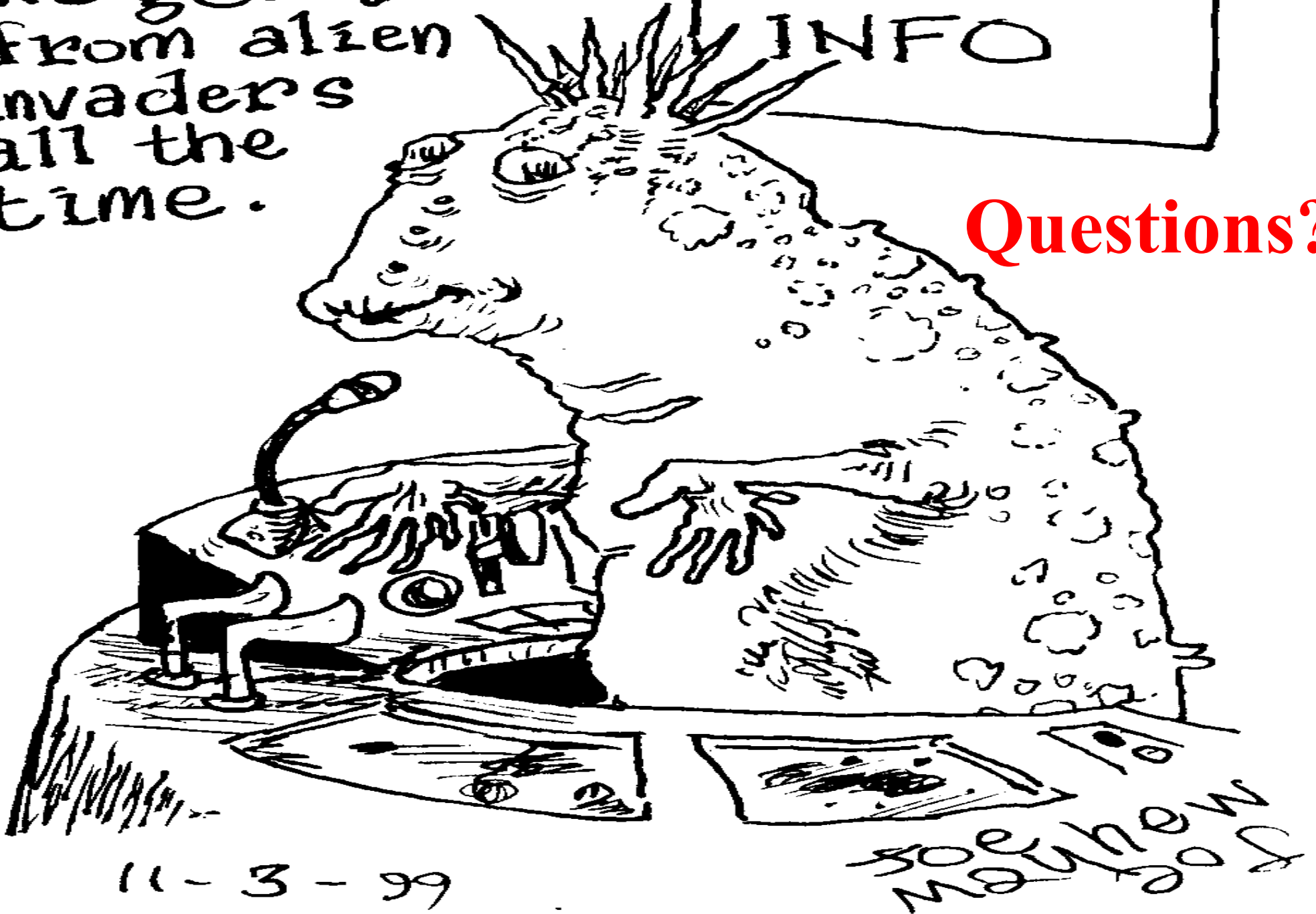
- Working::Daemon is still under development
- As of version 0.31, this does not work:
 - Logging
 - Cleaning up chroot
- But it does a lot of things right :)



No problem.
We get questions
from alien
invaders
all the
time.

TOURIST
INFO

Questions?



11-3-99

of
new
York
Times