SAWDOG 2.2.X DOCUMENTATION

CHRISTIAN GLOOR

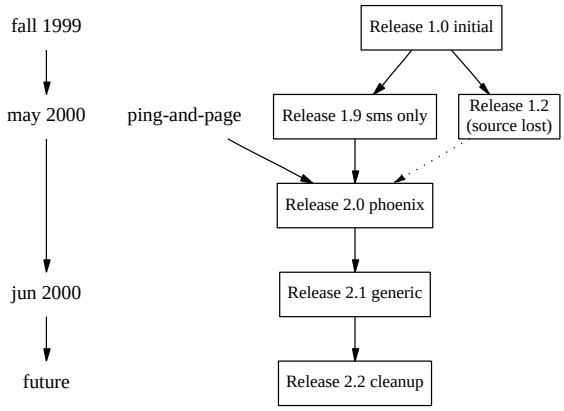
ABSTRACT. The sawdog script was initially written by Christian Gloor. About half a year later, it was adapted for use at *DIGICOMP AG*[1], a IT education company located in switzerland. Sawdog consists of a script which informs the sysops of some mission critical servers in the case of a failure. The script executes a given set of small executables, i.e. expect scripts, and if one executable fails, sends an email or a sms to the sysop.

SAWDOG - Simple Active Watch-DOG
Simple: less than 200 lines of code
Active: No need for daemons on the servers
SAWDOG - A collection of simple scripts, which informs in case of server outages

This collection of scripts is distributed under the terms of the GNU General Public License (GPL) Current maintainer: chgloor@digicomp.ch, don't hesitate to contact him if you have any questions.

INTRODUCTION

Sawdog consists of a script which informs the sysops of some mission critical servers in the case of a failure. The script executes a given set of small executables, i.e. expect scripts, and if one executable fails, sends an email or a sms to the sysop.



Changelog.

Fall 1999: 1.0.0 initial - chgloor@digicomp.ch.

- initial release for internal use only
- SMS and EMail notification

Winter 1999: 1.2.0 lost - source lost.

- Many new features

May 2000: 1.9.0 - michi@digicomp.ch.

- Changed for sms use only

Date: Copyright 1999-2000 by Christian Gloor.

May 2000: 2.0.0 phoenix - chgloor added additional features, inspired by ping-and-page [2].

- unknown state
- logfile
- locking

May 2000: 2.1.0 generic - chgloor, code cleanup, documentation.

- compact notification (only one message per server)
- aliases
- generic notification (not just sms or email)

May 2000: 2.2.0 speedup - chgloor, code cleanup.

- additional expect scripts
- 'required' feature

Installation

The main script is `sawdog.pl`. It can be placed everywhere on the filesystem. At the moment, the only configuration is done in the `sawdog.conf` file. However, if you don't keep this file in the current directory, you'll need to edit the `sawdog.pl` source file, because all the file locations are stored in the `sawdog.pl` file.

You can use crontab to start `sawdog.pl` every 5 minutes. I recommend the use of a simple start file, containing the commands `cd sawdog-dir`, `execute sawdog`.

Sawdog creates a logfile: `sawdog.log` and a status file: `sawdog.status`, if they don't exist, sawdog recreates them. Of course sawdog needs write permissions on the working directory. There are some predefined expect scripts for some well known services: `ftp`, `http`, `icmp`, `ssh`, `telnet`, `smb` and `notes`. You can easily create your own expect scripts, or use any other executable which returns a return code of zero if it succeeds. Don't forget to contribute your additions to the main code tree.

Files needed:

- `sawdog.pl` the main file
- `sawdog.conf` the config file
- `services/` this directory is populated by the expect scripts

Files recreated, if they don't exist:

- `sawdog.status` in this file the internal state is keept
- `sawdog.log` the logfile
- `sawdog.lock` the lockfile while it is running

Configuration

The main configuration file is `sawdog.conf`. It is a simple text file and contains a single line for each server.

Syntax: `server.domain.top (executable1[, executable2]) email[@domain.top]`

You can specify up to 5 executables. New to verision 2.1.x is the possibility to specify external notification methods.

```
.sms (echo '%message%' | /bin/mail sms\@smsgate.mydomain.com -s %user%)
```

There are 2 tokens which are replaced at execution time: `%message%` is substituted by the list of up/down ports, `%user%` by the username given in the configuration file.

If the IP name of a server is unhandy long, you can specify an alias. Separate the server name and the alias by a whitespace: `extranet.digicomp.ch CityWebSrv`. In the notification messages, the alias is used instead of the real name.

You can put an exclamation mark in front of a line. Sawdog stops emediately aborts if a server with an exclamation mark is not reachable. This way you can check if the server running sawdog has network connectivity.

Example. `sawdog.conf`

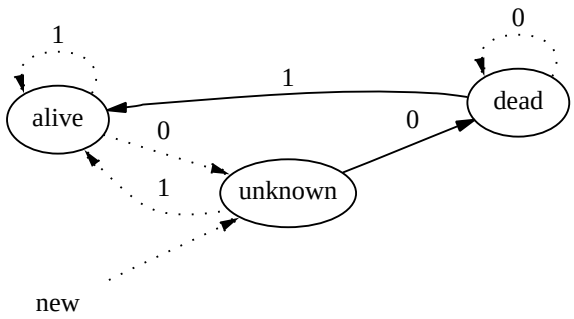
```
#notification methods
.default (echo '%message%' | /bin/mail %user% -s "sawdog message")
.sms (echo '%message%' | /bin/mail sms\@smsgate.mydomain.com -s %user%)

# internet
!www.ethz.ch (icmp) inet
!www.digicomp.ch (http, smtp) inet
one.gatekeeper.digicomp.ch (ssh, smtp) anet
two.gatekeeper.digicomp.ch (ssh, smtp) anet
extranet.digicomp.ch CityWebSrv (smtp, http, notes) notesteam
# housing
mordor.ch (ssh, smtp, telnet, http, ftp) chgloor
```

How it works

Sawdog consists of a script which informs the sysops of some mission critical servers in the case of a failure. The script executes a given set of small executables, i.e. expect scripts, and if one executable fails, sends an email or a sms to the sysop.

Sawdog knows three states for each service observed: *alive*, *unknown*, and *dead*. If a service is running, the service is marked as *alive* and the script does nothing until it's revoked next time. If, on the other hand, it recognizes that a service marked as *alive* is down, it marks the service as *unknown*. This is to avoid false alarms. Once in the *unknown* state, there are two possibilities: first, the script finds an active service at the next run, so it marks the servise as *alive* again. Second, if the service seems still not reachable, it has to suppose that the service is dead for sure. It marks the service as *dead,* and triggers notification. There's nothing the script can do at the moment. As soon as the script recognizes that the service is up again, it marks it as *alive* immediately and triggers a notification again.



To Do

- Create some more expect scripts for more ports.
- Build a nice package.
- Correct spelling and grammar errors in the readme files (my native language is german, not english. Don't hesitate to send any corrections).
- Version 3.0 will be a complete code rewrite.

References

[1] DIGICOMP AG: http://www.digicomp.ch/
[2] ping-and-page: http://expect.nist.gov/scripts/ping-and-page, Author unknown