An audit of all crontab entries on GOC machines

1 Introduction

All scripts executed periodically (referred to as a “cron” below) have been examined for potential ability to damage the system running it. It was found that the potential for damage can be arranged by the scope of the potential damage as follows:

- Crons that could damage all machines, discussed in section 2.
- Crons that could damage a virtual machine host or a physical machine running BDII, discussed in section 3.
- Service specific crons that could damage a single virtual machine, discussed in section 4.

In the following, paragraph headers are the name of a file (a cron) contained in /etc/cron.d on one or more machines. Subsection headers are the first few characters of the name of a machine running a specific service. All crons on all production machines are listed and breifly described.

2 Crons running on all machines

These crons have the potential to damage all machines, physical and virtual.

certsync Every hour, 36 minute offset. Use the yum server to update to the latest CA certificate RPM.

confsync-dyndns Every 5 minutes. Add GOC users’ dynamic DNS hosts to appropriate firewall rules.

confsync-rsync Every 5 minutes, 1 minute offset. Get any new confsync tarballs from the confsync server.

confsync-install Every 5 minutes, 2 minute offset. If confsync-rsync downloaded any tarballs, install the files from those tarballs.

ssh_hostkey_sync Every 5 minutes, 3 minute offset. Synchronize SSH public host keys with a server. Check to see whether the SSH public host key is new, and if it is, email it to a central address. The key is emailed with S/MIME encryption, using a certificate created just for this purpose; the key for decryption resides on the destination server.

crlsync-rsync Every 6 hours, 33 minute offset. Get the crlsync tarball from the crlsync server, if it’s new.

run_goc_update Once per minute. A utility script to propagate a change to all machines using confsync. It currently does nothing; it must be edited to perform some specific function.

sysstat Every 10 minutes. Run system activity accounting tool (/usr/lib64/sa/sa1) Also generate a system activity summary once per day.

3 Crons running only on physical machines

The physical machines “huey” and “ruckus” serve as hosts for several virtual machines. Damage to these machines could make the virtual machines unavailable.

make_mach_list sends to a GOC internal machine a list of virtual machines running on a physical machine. Invoked 38 minutes after every hour.

ipmiget Every minute. Run hardware monitoring tool, saving results to a temporary file that munin-node can read when the Munin server queries it about hardware status.

4 Crons running on specific machines

These crons run only on specific machines. These have the potential to make a specific service unavailable. All of these are specifically related to the service running on the machine. In cases where there are a pair of machines (myosg1, myosg2, for example) it has been verified both run the same set of crons.

4.1 BDII servers

These are physical machines running the BDII service. They also run the crons discussed in section 3.

bdii-proxy 4 times per day, offset 5 hours, 37 minutes, update environment and proxies.

4.2 data

data Backup the MySQL databases every 4 hours. Every 15 minutes report the status of the service running on this machine.

4.3 display

display Backup site_config.php every 4 hours. Every 15 minutes report the status of the service running on this machine.

osg_display.cron Every 5 minutes. Update display service.

4.4 jira

jira Every 3 hours, 30 minute offset, dump the latest jira.sql. Every hour backup all export files.

4.5 myosg

gip-validator 4 times per hour, offset one minute, cache MyOSG resource listing, etc. locally. 4 times per hour, 5 minute offset, launch the GIP validator. 4 times per hour 10 minute offset, launch the GIP validator for ITB.

myosg Every 15 minutes report the status of the service running on this machine. Every 4 hours back up config.php. Every minute update processed RSV data. Every 5 hours update VO data. Every 4 hours backup rsvprocess.conf.

test Every 30 minutes write the last 20 lines of various log files to a temporary location.

top-level-wlcg-bdii-monitor Once per hour, 12 minute offset, cache MyOSG data locally.

voms-monitor Every minute concatenate stdout and stderr from voms-monitor to a file in /var. Once per hour, offset 44 minutes concatenate stdout and stderr from vomes-monitor to a file in /var

4.6 oim

oim Once per hour, offset 24 minutes, dump oim.sql. At 4:00, generate the OIM daily report. At 16:00, generate the OIM service report. Every 15 minutes report the status of the service running on this machine.

4.7 rsv-client

rsv-client Currently does nothing.

4.8 rsv-process

rsvprocess Every minute do RSV processing of rapidly changing data. At 8:01 do RSV processing of slowly changing data. At 0:00 do daily RSV processing. At 4:00 backup rsvprocess.conf. At 5:00 dump optimize.sql. Every 15 minutes report the status of the service running on this machine.

4.9 ticket

ticket At 3:30 get groupticket for production instance. At 4:00 backup config.php. At 4:30 get groupticket for ITB intance. At 5:30 get groupticket for storage instance. Every 15 minutes report the status of the service running on this machine.

4.10 tx

tx At 4:45 dump tx database and backup to central server. Every 15 minutes report the status of the service running on this machine.

4.11 twiki

twiki At 0:00, generate mail notification, a TWiki provided script. At 0:00 generate web statistics. Once per day, garbage collection and cleanup. Every 4 hours, backup to GOC backup machine.

4.12 software and voms

These virtual machines have no additional crons.

5 Discussion

The crons of section 2 are of particular concern because of the possibility of damage to all services. In fact, the outage of July 15, 2010 involved confsync-install.

confsync-install and confsync-rsync while nominally running every 5 minutes actually do nothing unless a particular file changes on a central machine. A policy has been implemented forbidding changes to this file outside announced change windows. Likewise run_goc_update with its 1 minute period currently does nothing. Changes to this script are required for it to do something and such changes are also restricted to announced change windows.

ssh_hostkey_sync does nothing unless an ssh host key changes on some machine. This happens, for example, when a virtual machine is rebuilt following a failure. When needed it is essential, without this update processes cannot communicate to other machines because of the configuration of ssh on GOC machines (StrictHostKeyChecking yes). This update should be considered part of the process of restoring a service after failure. This cron is currently allowed to run as usual.

certsync (1 hour period) can potentially disable communication with external machines. CA RPM are updated rarely and a reversion procedure is in place. This cron is currently allowed to run as usual.

crlsync-rsync (6 hour period) does nothing unless a central file changes. Changes to this file are restricted by policy to announced change windows.

sysstat and confsync-dyndns have minimal potential for damage; these provide information for system monitoring and ease of use for GOC staff.

All crons in section 3 are currently allowed to run as usual. make_mach_list simply forwards a list of virtual machines running on a physical machine. ipmiget uses well known, system provided tools to report machine health and status.

All crons of section 4 are currently all allowed to run as usual. These all should be considered as part of the service running on a specific machine and are required by that service. Failure of any of these should be treated similarly to any other failure of a service.