Rudder - Bug #4498

Several issues with process management on Proxmox host (and container)

2014-02-22 01:30 - Alex Tkachenko

Status:	Rejected		
Priority:	N/A		
Assignee:			
Category:	System techniques		
Target version:	3.1.22		
Pull Request:		**Priority:**	57
Severity:	Critical - prevents main use of Rudder | no workaround | data loss | security	**Name check:**	
UX impact:		**Fix check:**	
User visibility:	Infrequent - complex configurations | third party integrations	**Regression:**	
Effort required:			

Description

I had my concerns regarding openVZ process management in cfengine, but after establishing the fact that Rudder uses special add-on patch I decided to give it a try and deployed rudder on several OpenVZ/Proxmox hosts and containers.

The first sign of trouble were two emails I've received at 5:05 am today and yesterday from one of the containers:

```
WARNING: No disable file detected and no CFEngine process neither. Relaunching CFEngine processes.
.. Done
```

I sorta ignored the first one, but having received the second at exactly same time I went to check on the matters. Initially I thought it is from cron, but it turned to be more complicated than that. The cron check is executed every 5 minutes, so why only 5:05? Also by looking at the outputs directory **both** on the container and the host I've noticed size differences (they are practically the same, so only one is presented here):

```
# ls -l *
...
-rw------- 1 root root 3578 Feb 21 04:46 cf_hostname__1392986764_Fri_Feb_21_04_46_04_2014_0x7fd054
c69700
-rw------- 1 root root 3578 Feb 21 04:50 cf_hostname__1392987029_Fri_Feb_21_04_50_29_2014_0x7fd054
c69700
-rw------- 1 root root 3578 Feb 21 04:55 cf_hostname__1392987354_Fri_Feb_21_04_55_54_2014_0x7fd054
c69700
-rw------- 1 root root  946 Feb 21 05:01 cf_hostname__1392987679_Fri_Feb_21_05_01_19_2014_0x7fd054
c69700
-rw------- 1 root root 3578 Feb 21 05:05 cf_hostname__1392987947_Fri_Feb_21_05_05_47_2014_0x7f5e78
1c4700
-rw------- 1 root root 3578 Feb 21 05:11 cf_hostname__1392988272_Fri_Feb_21_05_11_12_2014_0x7f5e78
1c4700
...
```

The file with a different size has a content:

```
2014-02-21T05:01:21-0800   notice: R: @@Common@@log_info@@hasPolicyServer-root@@common-root@@109@@
common@@StartRun@@2014-02-21 05:01:21-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#Start execution
2014-02-21T05:01:21-0800   notice: R: @@Common@@result_success@@hasPolicyServer-root@@common-root@
@109@@Update@@None@@2014-02-21 05:01:21-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#Rudder policy
, tools and ncf instance are already up to date. No action required.
2014-02-21T05:01:21-0800   notice: R: @@Common@@result_success@@hasPolicyServer-root@@common-root@
@109@@Security parameters@@None@@2014-02-21 05:01:21-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#
The internal environment security is acceptable
2014-02-21T05:01:21-0800   notice: R: @@Common@@result_success@@hasPolicyServer-root@@common-root@
@109@@Red Button@@None@@2014-02-21 05:01:21-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#Red Butto
n is not in effect, continuing as normal...
```

Normally this should be followed by a line

```
2014-02-21T05:05:49-0800   notice: R: @@Common@@result_success@@hasPolicyServer-root@@common-root@
@109@@Process checking@@None@@2014-02-21 05:05:49-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#The
re is an acceptable number of CFEngine processes running on the machine
```

but it cuts right there as if the process was terminated. I suspected that there might be some problems with getting the process list, but as vzps should be supported - what's the problem? OK, here comes the first problem. The patch, as presented here: https://github.com/Normation/rudder-packages/blob/da3db60a6ffd01ce7c581c575321b866c3c1db61/rudder-agent/SOURCES/add-support-of-openVZ.patch lists the following parameters for vzps:

```
+    "-o user,pid,ppid,pgid,pcpu,pmem,vsz,pri,rss,nlwp,stime,time,args", /* vz with vzps */
```

Unfortunately, this does not work on Proxmox, as they for whatever reason ship quite an old version, based on procps v2, which does not support nlwp option. They support thcount option instead, and the main page says that

```
nlwp        NLWP    number of lwps (threads) in the process. (alias thcount).
```

So, basically, replacing nlwp with thcount would work for both the old and the new case. CFengine has a similar issue, see https://cfengine.com/dev/issues/3395.

I do not know what happens exactly, but either the vzps exit status is misinterpreted or it reverts to using a regular ps - but apparently the host starts shooting off container's processes (which are restarted later because of those 5-minute-apart cron checks - hence the email). Oh, and actually a magic 5:05 - apparently this is the time when cfengine processes are forcibly restarted by the policy (see common/1.0/process_matching.cf). Only at 5:05 it actually tries to stop the processes - and how the kill is implemented is not clear - but evidently something goes wrong.

To add an insult to an injury, I have found that on my host the cf-execd process is not running for several hours. It is 16:11 now, and the last non-empty file in the outputs directory timestamped with 11:15:

```
...
2014-02-21T11:15:41-0800   notice: R: @@Common@@result_repaired@@hasPolicyServer-root@@common-root
@@109@@Process checking@@None@@2014-02-21 11:15:41-08:00##65618d90-0e20-47d2-bf1f-693305dd51bc@#Wa
rning, more than 2 cf-execd processes were detected. They have been sent a graceful termination si
gnal.
...
```

The next file is empty and timestamped by 11:16:

```
# ls -l *_Fri_Feb_21_11_16_06_2014_0x7f5e781c4700
-rw------- 1 root root 0 Feb 21 11:16 cf_hostname__1393010166_Fri_Feb_21_11_16_06_2014_0x7f5e781c4
700
```

And nothing beyond that until now. Again, something went wrong and the process apparently killed itself.

So why it is not being restarted by cron? This is another issue. The cron file (generated by the policy) contains the line

```
0,5,10,15,20,25,30,35,40,45,50,55 * * * * root if [ -e /opt/rudder/bin/check-rudder-agent ]; then
/opt/rudder/bin/check-rudder-agent; else if [ ! -e /opt/rudder/etc/disable-agent -a `ps -efww | gr
ep -E "(cf-execd|cf-agent)" | grep -E "/var/rudder/cfengine-community/bin/(cf-execd|cf-agent)" | g
rep -v grep | wc -l` -eq 0 ]; then /var/rudder/cfengine-community/bin/cf-agent -f failsafe.cf >/de
v/null 2>&1 && /var/rudder/cfengine-community/bin/cf-agent >/dev/null 2>&1; if [ $? != 0 ]; then i
f [ -f /opt/rudder/etc/rudder-restart-message.txt ]; then cat /opt/rudder/etc/rudder-restart-messa
ge.txt; else echo "Rudder agent was unable to restart on $(hostname)."; fi; fi; fi; fi
```

So it tries to use check-rudder-agent script, but if it is not available it uses the plain ps (which in this case would yield the incorrect result). But we **do** have the script - however, it turns out that it uses plain ps as well:

```
# If no disable file AND no process of CFEngine from Rudder, then relaunch cf-agent with a failsaf
e first
# But this is applied only on servers or nodes already initialized (policy server set)
if [ ! -e ${CFE_DISABLE_FILE} -a `ps -efww | grep -E "(cf-execd|cf-agent)" | grep -E "${CFE_BIN_DI
```

```
R}/(cf-execd|cf-agent)" | grep -v grep | wc -l` -eq 0 -a -f ${CFE_DIR}/policy_server.dat ]; then
  echo -n "WARNING: No disable file detected and no CFEngine process neither. Relaunching CFEngine
 processes..."
  ${CFE_BIN_DIR}/cf-agent -f failsafe.cf >/dev/null 2>&1
  ${CFE_BIN_DIR}/cf-agent >/dev/null 2>&1
  echo " Done"
fi
```

As the containers still have the cf-execd running, the script never detects that it needs to relaunch the process on the mothership.

I am restarting the service manually for now (as it is the end of the working week :) - we shall see how it is working over the weekend and will continue on Monday.

Subtasks:

Bug # 4534: vzps implementation in cfengine doesn't work with Promox (branch 2.11)	**Rejected**
Bug # 4535: vzps implementation in cfengine doesn't work with Promox (branch 2.6)	**Rejected**
Bug # 5467: implement openvz support for rudder 2.11	**Released**

Related issues:

Related to Rudder - Bug #4499: Rudder init script kill all agent on Open VZ (...	**Released**	**2014-02-23**
Related to Rudder - Bug #4509: vzps is never used on vzps system	**Rejected**	**2014-02-25**
Related to Rudder - Bug #4517: vzps is never used on vzps system (branch 2.8)	**Rejected**	**2014-02-25**
Related to Rudder - Bug #7189: issues with process management on physical hos...	**Released**	**2015-09-12**
Related to Rudder - Bug #7381: Process management issues on nodes hosting LXC...	**Released**	
Related to Rudder - Bug #7423: If using proxmox, process management fails due...	**Released**	**2015-12-07**

History

#1 - 2014-02-22 01:36 - Alex Tkachenko

I've also checked the service startup script. Unfortunately it looks like this one may need some fixin' too:

```
...
stop_daemons() {
        RET=0

        for daemon in ${DAEMONS}; do
                # Stop message
                message "info" "[INFO] Halting CFEngine Community ${CFENGINE_COMMUNITY_NAME[$daemon]}..."

                # Presence of PID file
                # If the pid file is not readable or is empty, kill all process by name
                if [ ! -r ${CFENGINE_COMMUNITY_PID_FILE[$daemon]} -o ! -s ${CFENGINE_COMMUNITY_PID_FILE[$daemo
n]} ]
                then
                        message "info" "[INFO] can't read PID file, not stopping ${CFENGINE_COMMUNITY_NAME[$da
emon]}"
                        RET=1
                else
...
```

OK, at least here it will not try to kill the process if the pid is unknown. But going further, in forcestop_daemons:

```
                # Presence of PID file
                # If the pid file is not readable or is empty, kill all process by name
                if [ ! -r ${CFENGINE_COMMUNITY_PID_FILE[$daemon]} -o ! -s ${CFENGINE_COMMUNITY_PID_FILE[$daemo
n]} ]
                then
                        # Try a killall
                        /usr/bin/killall -KILL ${CFENGINE_COMMUNITY_BIN[$daemon]}
```

This would definitely kill them all, not only on the mothership, but on the containers as well.

#2 - 2014-02-22 01:40 - Alex Tkachenko

Damn, I did not even leave the building and the process is already killed :)
I should have three cf-execd processes running, but I have only one, which means that the very first check killed one on the host and on one of the containers.

#3 - 2014-02-23 09:38 - Nicolas CHARLES

- Category set to System techniques

- Priority changed from N/A to 1

Hi Alex,

These are indeed quite awfull bugs :(

Please note that the next minor release of Rudder will have a corrected check_rudder-agent (see https://github.com/Normation/rudder-packages/blob/branches/rudder/2.9/rudder-agent/SOURCES/check-rudder-agent)

I don't have sufficient knowledge of the ps command line parameters to know the impact of the change from nlwp to thcount; especially on "standart" systems

And I'm opening another ticket for the init script.

Thank you for the bug report, and sorry for the bugs

#4 - 2014-02-24 19:59 - Alex Tkachenko

The check-rudder-agent script looks good to me in the quoted revision (although I assume you're gonna include the vzps.py in the next package revision, as I do not seem to have it in my 2.9.2 installation). This change should be sufficient to recover from the agent death on an openVZ host - however the policy-enforced cron line still uses plain ps unconditionally, as a backup - maybe it should be changed too? For the sake of simplicity it could use a rudder-provided script, instead of encoding the check logic into a cron one-liner.

The ps arguments change is **necessary** because in the present form it will **not** work:

```
# vzps -o user,pid,ppid,pgid,pcpu,pmem,vsz,pri,rss,nlwp,stime,time,args
ps: error: Unknown user-defined format specifier.
usage: vzps -[Unix98 options]
       vzps [BSD-style options]
       vzps --[GNU-style long options]
       vzps --help for a command summary
# vzps -o user,pid,ppid,pgid,pcpu,pmem,vsz,pri,rss,thcount,stime,time,args | head -2
USER       PID  PPID  PGID %CPU %MEM   VSZ PRI  RSS THCNT STIME TIME COMMAND
root     51843 51841 51843  0.0  0.0 19440  19 2260 - Feb21 00:00:00 bash -rcfile .bashrc
```

I have checked all the linux installation I could reach, and on all systems thcount is an alias for nlwp, so this change is pretty safe. Another alternative would be to force the vendor of Proxmox to update their shipped version of vzps, or provide a (debian) package in some repository to replace the shipped version. The OpenVZ vendor though, offers this ancient version from their site (http://download.openvz.org/debian-systs/pool/openvz/v/vzprocps/) so it may be not up to the Proxmox folks to address it. As a matter of fact, the RPM, suggested at http://openvz.org/Processes_scope_and_visibility, at http://download.openvz.org/contrib/utils/ is also of the same version (2.0.11), so I guess that the patch should include thcount instead of nlwp for **all** openvz instances.

And that is for the case when **it is actually used**, because as I found this weekend, this is not the case.
To check if it is actually called, I created a simple script, which would log the arguments and then call the real vzps. To my surprise, after the agent was fired (and killed), the log file remained empty. Puzzled, I run cf-agent -Kv, and here is what I have found.

In the output, the hard classes line does include the virt_host_vz_vzps class, but this is **the only** line referencing to vzps.

```
# grep vzps agent-kv-output.txt
2014-02-22T09:05:26-0800  verbose: Discovered hard classes: 10_202_104_204 127_0_0_1 4_cpus 64_bit Day22 February GMT_Hr17 Hr09 Hr09_Q1 Lcycle_1 Min05 Min05_10 Morning PK_MD5_c36ff4f8b048b41fec9050eb58ff8aaf Q1 Saturday Yr2014 agent any carrieriq_com cfengine cfengine_3 cfengine_3_5 cfengine_3_5_2 com community_edition compiled_on_linux_gnu debian debian_6 debian_6_0 fe80__1 fe80__21e_c9ff_fe54_d64 fe80__3819_c8ff_fef7_60c9 fe80__60dc_efff_fef3_10dd fe80__b432_65ff_fec7_68fd have_aptitude inform_mode ipv4_10 ipv4_10_202 ipv4_10_202_104 ipv4_10_202_104_204 ipv4_127 ipv4_127_0 ipv4_127_0_0 ipv4_127_0_0_1 linux linux_2_6_32_19_pve linux_x86_64 linux_x86_64_2_6_32_19_pve linux_x86_64_2_6_32_19_pve__1_SMP_Mon_Mar_18_06_41_32_CET_2013 localhost localhost_localdomain mac_00_1e_c9_54_0d_64 net_iface_lo net_iface_vmbr0 pvelocalhost sjc1_ops_proxmox01 sjc1_ops_proxmox01_carrieriq_com verbose_mode virt_host_vz virt_host_vz_vzps x86_64
```

According to the patch, this should have trigger the use of vzps-specific entries, but further down, when it comes to process checking,
we see:

```
# grep "process table" agent-kv-output.txt
2014-02-22T09:05:27-0800  verbose: Observe process table with /bin/ps -eo user,pid,ppid,pgid,pcpu,pmem,vsz,ni,rss,nlwp,stime,time,args
2014-02-22T09:05:27-0800  verbose: Observe process table with /bin/ps -eo user,pid,ppid,pgid,pcpu,pmem,vsz,ni,rss,nlwp,stime,time,args
2014-02-22T09:05:27-0800  verbose: Observe process table with /bin/ps -eo user,pid,ppid,pgid,pcpu,pmem,vsz,ni,rss,nlwp,stime,time,args
2014-02-22T09:05:27-0800  verbose: Observe process table with /bin/ps -eo user,pid,ppid,pgid,pcpu,pmem,vsz,ni,rss,nlwp,stime,time,args
```

which corresponds to the linux class line.

I now wonder, how that table is being used, as we have two classes which could be used to select entries from that table, specifically linux and virt_host_vz_vzps - so what are the precedence rules?

#5 - 2014-02-24 20:57 - Alex Tkachenko

Apparently the vzps-specific entries are never used, because VSYSTEMHARDCLASS is not affected when virt_host_vz_vzps is detected.
I found that it is defined in GetNameInfo3, but the OpenVZ_Detect function introduced in the patch only **adds** the hard class to the context later on, so when it comes to process table inspection in LoadProcessTable, the "linux" selector is effecting the actual line.

```
snprintf(pscomm, CF_MAXLINKSIZE, "%s %s", VPSCOMM[VSYSTEMHARDCLASS], psopts);
```

#6 - 2014-02-25 09:11 - Nicolas CHARLES

Alex,

This is a really good catch. Indeed the detection of VZ is made after the definition of the VSYSTEMHARDCLASS, nulling all the effect of the code specific for VZ
I'm opening a separate ticket for this issue

#7 - 2014-02-25 18:06 - Alex Tkachenko

Well, it's not entirely in vain, as the classes are defined and could be used from within the policy, but of course internal processes promises won't work (but again, the classes could be used to make those promises conditional).

It is possible to workaround the problem on the mothership by enabling certain features in the kernel to hide container's processes from the host, but it is said that this feature is incompatible with the live migration and checkpointing.

So the bottom line - we need this feature operational :)

Please let me know how could I help to expedite the matter.

#8 - 2015-09-16 17:41 - Nicolas CHARLES

- Related to Bug #7189: issues with process management on physical hosting LXC containers added

#9 - 2015-11-06 14:21 - Alexis Mousset

- Related to Bug #7381: Process management issues on nodes hosting LXC containers added

#10 - 2015-11-25 23:31 - Jonathan CLARKE

- Related to Bug #7423: If using proxmox, process management fails due to bad options used on vzps added

#11 - 2016-05-11 15:21 - Benoît PECCATTE

- Target version set to 2.11.21

#12 - 2016-05-24 23:48 - Vincent MEMBRÉ

- Target version changed from 2.11.21 to 2.11.22

#13 - 2016-06-02 18:22 - Vincent MEMBRÉ

- Target version changed from 2.11.22 to 2.11.23

#14 - 2016-07-29 15:17 - Vincent MEMBRÉ

- Target version changed from 2.11.23 to 2.11.24

#15 - 2016-08-27 12:04 - Vincent MEMBRÉ

- Target version changed from 2.11.24 to 308

#16 - 2016-09-12 12:00 - Vincent MEMBRÉ

- Target version changed from 308 to 3.1.14

#17 - 2016-09-28 22:52 - Vincent MEMBRÉ

- Target version changed from 3.1.14 to 3.1.15

#18 - 2016-10-03 12:16 - Vincent MEMBRÉ

- Target version changed from 3.1.15 to 3.1.16

#19 - 2016-10-13 17:03 - Vincent MEMBRÉ

- Target version changed from 3.1.16 to 3.1.17

#20 - 2016-12-05 15:21 - Vincent MEMBRÉ

- Target version changed from 3.1.17 to 3.1.18

#21 - 2017-02-17 23:22 - Vincent MEMBRÉ

- Target version changed from 3.1.18 to 3.1.19

#22 - 2017-04-04 09:43 - François ARMAND

- Severity set to Critical - prevents main use of Rudder | no workaround | data loss | security

- User visibility set to Infrequent - complex configurations | third party integrations

- Priority set to 0

#23 - 2017-04-14 17:09 - Vincent MEMBRÉ

- Target version changed from 3.1.19 to 3.1.20

#24 - 2017-05-18 23:13 - Vincent MEMBRÉ

- Target version changed from 3.1.20 to 3.1.21

#25 - 2017-05-22 17:16 - François ARMAND

- Priority changed from 0 to 42

#26 - 2017-06-15 10:52 - Vincent MEMBRÉ

- Target version changed from 3.1.21 to 3.1.22

#27 - 2017-06-26 12:35 - Benoît PECCATTE

- Priority changed from 42 to 43

#28 - 2017-06-26 18:43 - Benoît PECCATTE

- Priority changed from 43 to 56

#29 - 2017-08-03 16:34 - Alexis Mousset

- Status changed from New to Rejected

- Priority changed from 56 to 57

Support for rudder-agent in both host and containers is fixed now (all CFEngine patches were merged upstream, our init and health-check scripts were fixed), and I successfully use it with Rudder 4.1 and Proxmox 4.4 (now LXC based).

I's closing this issue, feel free to open a new one if problems of this type still happen.