Pulp - Issue #7663

SELinux policies not being applied when using RPM based installation

10/07/2020 01:16 PM - spredzy

Status:	CLOSED - CURRENTRELEASE	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	mdepaulo@redhat.com	**Estimated time:**	0:00 hour
Category:	Installer		
Sprint/Milestone:	3.8.0		
Severity:	4. Urgent	**Groomed:**	No
Version:		**Sprint Candidate:**	No
Platform Release:		**Tags:**	SELinux
OS:		**Sprint:**	
Triaged:	Yes	**Quarter:**	

Description

As a user I am deploying pulpcore using the pulp_installer (3.7.1) and the upstream packages located at https://yum.theforeman.org/pulpcore/3.7/el7/x86_64/

After installation the services are started as unconfined_service_t rather than pulpcore_t (as per the pulpcore-selinux definition https://github.com/pulp/pulpcore-selinux/blob/master/pulpcore.te#L8)

```
[root@localhost vagrant]# rpm -qa | grep pulpcore-selinux
pulpcore-selinux-1.1.1-1.el7.x86_64

[root@localhost vagrant]# semodule -l | grep pulp
pulpcore        1.1.1
pulpcore_port   1.1.1
pulpcore_rhsmcertd      1.1.1

[root@localhost vagrant]# ps -Z fauxwww | grep pulpcore-api
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 3533 0.0  0.1 12532 984 pts/1 S+ 11:07
  0:00                          \_ grep --color=auto pulpcore-api
system_u:system_r:unconfined_service_t:s0 pulp 2986 0.0  1.0 122984 5220 ?     Ss   09:29   0:02 /
usr/bin/python3 /usr/bin/gunicorn pulpcore.app.wsgi:application --bind unix:/var/run/pulpcore-api/
pulpcore-api.sock --workers 4 --access-logfile -
system_u:system_r:unconfined_service_t:s0 pulp 2989 0.0  6.9 276252 34836 ?    S    09:29   0:01
\_ /usr/bin/python3 /usr/bin/gunicorn pulpcore.app.wsgi:application --bind unix:/var/run/pulpcore-
api/pulpcore-api.sock --workers 4 --access-logfile -
system_u:system_r:unconfined_service_t:s0 pulp 2991 0.0  6.9 276252 34500 ?    S    09:29   0:01
\_ /usr/bin/python3 /usr/bin/gunicorn pulpcore.app.wsgi:application --bind unix:/var/run/pulpcore-
api/pulpcore-api.sock --workers 4 --access-logfile -
system_u:system_r:unconfined_service_t:s0 pulp 2992 0.0  6.5 276252 32848 ?    S    09:29   0:01
\_ /usr/bin/python3 /usr/bin/gunicorn pulpcore.app.wsgi:application --bind unix:/var/run/pulpcore-
api/pulpcore-api.sock --workers 4 --access-logfile -
system_u:system_r:unconfined_service_t:s0 pulp 2995 0.0  6.1 276344 30620 ?    S    09:29   0:01
\_ /usr/bin/python3 /usr/bin/gunicorn pulpcore.app.wsgi:application --bind unix:/var/run/pulpcore-
api/pulpcore-api.sock --workers 4 --access-logfile -
[root@localhost vagrant]#
```

Expected behavior is to have everything properly labeled

Related issues:

Related to Pulp - Issue #7667: pulp_installer should use the new libexec SELi...	**CLOSED - CURRENTRELEASE**

History

#1 - 10/07/2020 01:43 PM - bmbouter

The policy expects the rq and gunicorn services to be started from their binaries in either of these paths:
https://github.com/pulp/pulpcore-selinux/blob/master/pulpcore.fc#L11-L20

I see in the report the path /usr/bin/python3 /usr/bin/gunicorn so maybe that has something to do with them not receiving the right labels?

#2 - 10/07/2020 02:07 PM - spredzy

This is not something we control from an install perspective as a user

```
[root@localhost ansible-tower-setup-3.8.0-0.git20201006.37b3cc7]# systemctl cat pulpcore-api
# /usr/lib/systemd/system/pulpcore-api.service
[Unit]
Description=Pulp WSGI Server
After=network-online.target
Wants=network-online.target

[Service]
Environment="DJANGO_SETTINGS_MODULE=pulpcore.app.settings"
Environment="PULP_SETTINGS=/etc/pulp/settings.py"
Environment="PATH=/usr/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin"
Environment="LD_LIBRARY_PATH=/opt/rh/rh-postgresql10/root/usr/lib64"
User=pulp
Group=pulp
PIDFile=/run/pulpcore-api.pid
RuntimeDirectory=pulpcore-api
ExecStart=/usr/bin/gunicorn pulpcore.app.wsgi:application \
          --bind 'unix:/var/run/pulpcore-api/pulpcore-api.sock' \
          --workers 4 \
          --access-logfile -

# This provides reconnect support for PostgreSQL and Redis. Without reconnect support, if either
# is not available at startup or becomes disconnected, this process will die and not respawn.
Restart=always
RestartSec=3

# Enables tracking of process resource consumption at a unit & cgroup level.
CPUAccounting=yes
MemoryAccounting=yes

# This directive is set to an absolute path in other Pulp units. Using an
# absolute path is an abuse of the directive, as it should be a relative path,
# not an absolute path. PIDFile is now used to ensure that PID files are laid
# out in a standard way. If this directive had any other effects, it is better
# to use the correct directive than to uncomment this.
# WorkingDirectory=/var/run/pulpcore-api/

[Install]
WantedBy=multi-user.target
```

Now if you are wondering where the /usr/bin/python3 comes from in the original ticket, its the shebang in /usr/bin/gunicorn

```
[root@localhost ansible-tower-setup-3.8.0-0.git20201006.37b3cc7]# cat /usr/bin/gunicorn
#!/usr/bin/python3
# EASY-INSTALL-ENTRY-SCRIPT: 'gunicorn==20.0.4','console_scripts','gunicorn'
__requires__ = 'gunicorn==20.0.4'
import re
import sys
from pkg_resources import load_entry_point

if __name__ == '__main__':
    sys.argv[0] = re.sub(r'(-script\.pyw?|\.exe)?$', '', sys.argv[0])
    sys.exit(
        load_entry_point('gunicorn==20.0.4', 'console_scripts', 'gunicorn')()
    )
```

One can run ln s /usr/bin/python3 /usr/bin/python-pulp and update the shebang of /usr/bin/gunicorn like that

```
[root@localhost ansible-tower-setup-3.8.0-0.git20201006.37b3cc7]# cat /usr/bin/gunicorn
#!/usr/bin/python-pulp
# EASY-INSTALL-ENTRY-SCRIPT: 'gunicorn==20.0.4','console_scripts','gunicorn'
__requires__ = 'gunicorn==20.0.4'
import re
import sys
from pkg_resources import load_entry_point

if __name__ == '__main__':
    sys.argv[0] = re.sub(r'(-script\.pyw?|\.exe)?$', '', sys.argv[0])
    sys.exit(
```

```
        load_entry_point('gunicorn==20.0.4', 'console_scripts', 'gunicorn')()
    )
```

to be able to see the result

```
root      4190  0.0  0.1  12528   980 pts/0    S+   12:06   0:00  |                       \_ grep --color=auto
 pulpcore-api
pulp      4178  2.4  1.2 122984  6228 ?        Ss   12:06   0:00 /usr/bin/python-pulp /usr/bin/gunicorn pulpco
re.app.wsgi:application --bind unix:/var/run/pulpcore-api/pulpcore-api.sock --workers 4 --access-logfile -
pulp      4181 22.8 16.3 276284 81560 ?        S    12:06   0:01  \_ /usr/bin/python-pulp /usr/bin/gunicorn pu
lpcore.app.wsgi:application --bind unix:/var/run/pulpcore-api/pulpcore-api.sock --workers 4 --access-logfile -
pulp      4183 22.5 16.3 276284 81680 ?        S    12:06   0:01  \_ /usr/bin/python-pulp /usr/bin/gunicorn pu
lpcore.app.wsgi:application --bind unix:/var/run/pulpcore-api/pulpcore-api.sock --workers 4 --access-logfile -
pulp      4185 22.3 16.3 276284 81680 ?        S    12:06   0:01  \_ /usr/bin/python-pulp /usr/bin/gunicorn pu
lpcore.app.wsgi:application --bind unix:/var/run/pulpcore-api/pulpcore-api.sock --workers 4 --access-logfile -
pulp      4186 22.3 16.3 276308 81672 ?        S    12:06   0:01  \_ /usr/bin/python-pulp /usr/bin/gunicorn pu
lpcore.app.wsgi:application --bind unix:/var/run/pulpcore-api/pulpcore-api.sock --workers 4 --access-logfile -
```

#3 - 10/07/2020 02:40 PM - ekohl

What's the output of

```
ls -lZ /usr/bin/{gunicorn,rq}
```

I suspect they may not have been labelled correctly.

#4 - 10/07/2020 03:17 PM - ekohl

Summarizing the discussion on IRC:

```
# ls -lZ /usr/bin/gunicorn /usr/bin/rq
-rwxr-xr-x. root root system_u:object_r:bin_t:s0       /usr/bin/gunicorn
-rwxr-xr-x. root root system_u:object_r:bin_t:s0       /usr/bin/rq
```

A restorecon did reset them.

The problem is https://github.com/theforeman/pulpcore-packaging/blob/faabec8978c1cd9166da3fb715b7a97b3080c19c/packages/pulpcore-selinux/pulpcore-selinux.spec#L60:

```
/sbin/fixfiles -R python3-pulpcore restore || :
```

This restores the contexts on python3-pulpcore, but /usr/bin/{gunicorn,rq} are owned by python3-gunicorn and /usr/bin/rq.

https://github.com/theforeman/pulpcore-packaging/pull/37 is an effort to introduce wrappers. That's probably the best place to address this.

#5 - 10/07/2020 04:40 PM - mdepaulo@redhat.com

- Status changed from NEW to MODIFIED

#6 - 10/07/2020 04:43 PM - mdepaulo@redhat.com

- Status changed from MODIFIED to NEW

#7 - 10/07/2020 04:46 PM - mdepaulo@redhat.com

ekohl is correct, I want the wrappers to be the permanent fix. I want to work with them on those.

If there is an urgent need, pulp_installer can do 1 of the following temporary fixes:

1. Have the installer create the wrappers; identical to what the packages create.
2. Have the installer run fixfiles on those paths. It's already running fixfiles as part of the SELinux support for pip mode anyway, #7574.

EDIT: Only # 2 will work at this very moment, because the pulpcore-selinux RPM package needs to be updated from 1.1.1 to 1.1.2 for # 1 to work properly: https://github.com/pulp/pulpcore-selinux/releases/tag/1.1.2

#8 - 10/07/2020 05:13 PM - mdepaulo@redhat.com

- Assignee set to mdepaulo@redhat.com

#9 - 10/07/2020 07:20 PM - mdepaulo@redhat.com

- Related to Issue #7667: pulp_installer should use the new libexec SELinux wrappers from the RPM packages added

#10 - 10/07/2020 07:52 PM - mdepaulo@redhat.com

- Triaged changed from No to Yes

#11 - 10/07/2020 09:10 PM - mdepaulo@redhat.com

Per our IRC convo, I'm letting the updated packages get released. I just reviewed: https://github.com/theforeman/pulpcore-packaging/pull/38

#12 - 10/09/2020 01:09 PM - ekohl

By now pulpcore-selinux 1.1.3 has been released which contains the correct context. python3-pulpcore-3.7.1-2 contains the libexec wrappers. For convenience, I've submitted https://github.com/theforeman/pulpcore-packaging/pull/41 as a cherry pick to 3.6. That makes life easier in the Foreman Installer where the module still primarily 3.6 (since 3.7 was incomplete until yesterday).

#13 - 10/20/2020 09:20 PM - mdepaulo@redhat.com

- Status changed from NEW to CLOSED - CURRENTRELEASE

- Sprint/Milestone set to 3.8.0

Closing as the combined fix is part of the RPM packaging, and the pulpcore-selinux policy releases 1.1.3 / 1.1.2 .