Manage SELinux Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

Session - 36 Agenda:

1. Manage SELinux Using Ansible

SELinux:

SELinux, Security Enhanced Linux, is a security architecture for Linux. It controls access to applications, files and processes in a Linux system. It uses security policies that define what can or cannot be accessed. For example, if a process or application (termed as a subject) wants to access an object such as a file, SELinux checks against the corresponding rules and determine whether to allow or to deny the access.

How does SELinux Works?

By default, SELinux uses 'targeted policy' and is always **'enforcing'**. Targeted means that only specific (targeted) processes are protected by SELinux. Enforcing status means that SELinux policies are applied accordingly and operations are logged. You can change these settings under /etc/selinux/config file. Other status options are:

Permissive – SElinux rules are not applied but operations are logged in case there is a breach.
Disabled – SELinux policies not applicable

SElinux uses labelling and enforcement. Processes, files etc are labelled with a SELinux context. Files and directories have their labels stored as extended attributes on the filesystem while processes and ports labels are managed by the kernel. Labelling group the resources together such that those with the same labels access each other but are not allowed to access those with different labels. You can work around enabling access for example by changing the SELinux context of the file or process or changing the Boolean status. Booleans are basically the on/off settings of SELinux.

Managing SELinux:

You can manually manage SELinux on your Linux System by running the appropriate commands directly on your terminal. Most Linux packages needed for SELinux management are installed by default, but you may need to manually install others for your use. Some packages installed by default include:

policycoreutils provides utilities such as restorecon, secon, setfiles, semodule, load_policy, and setsebool, for operating and managing SELinux.
selinux-policy provides a basic directory structure, the selinux-policy.conf file, and RPM macros.
selinux-policy-targeted provides the SELinux targeted policy.
libselinux – provides an API for SELinux applications.
libselinux-utils provides the avcstat, getenforce, getsebool, matchpathcon, selinuxconlist, selinuxdefcon, selinuxenabled, and setenforce utilities.
libselinux-python provides Python bindings for developing SELinux applications.
Some optional packages that you may need to manually install are as below. Simply run ' sudo yum install <package-name>'
selinux-policy-devel provides utilities for creating a custom SELinux policy and policy modules.
selinux-policy-doc provides manual pages that describe how to configure SELinux altogether with various services.
selinux-policy-mls provides the MLS (Multi-Level Security) SELinux policy.
Settroubleshot and setroubleshoot-server translates denial messages, produced when access is denied by SELinux, into detailed descriptions that can be viewed with the sealert utility, also provided in this package.

Manage SELinux Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

mcstrans translates levels, such as s0-s0:c0.c1023, to a form that is easier to read, such as SystemLow-SystemHigh.
policycoreutils-python provides utilities such as semanage, audit2allow, audit2why, and chcat, for operating and managing SELinux.
policycoreutils-gui provides system-config-selinux, a graphical utility for managing SELinux.

 # sestatus
 # getenforce
 # setenforce 0
 # getenforce
 # sestatus
 # vim /etc/selinux/config
 # sestatus
 # ls -lZ /etc/redhat-release
 # ps -efZ | grep ssh
 # id
 # getsebool -a
 # getsebool <Boolean>
 # setsebool <Boolean> <on|off>
 # setsebool -P <Boolean> <1|0>
 # semanage port -l
 # semanage port -l | grep ssh
 # touch /tmp/file.txt
 # ls -lZ /tmp/file.txt
 # chcon -t httpd_sys_content_t /tmp/file.txt
 # ls -lZ /tmp/file.txt
 # semanage fcontext -l | grep /tmp/file.txt
 # semanage fcontext -a -t user_tmp_t /tmp/file.txt
 # semanage fcontext -l | grep /tmp/file.txt
 # restorecon -vR /tmp/file.txt
 # ls -lZ /tmp/file.txt

Managing SELinux with Ansible:
We can manage the SELinux related tasks on the managed nodes by using:
1. Ansible Ad-Hoc Commands
2. Ansible Playbooks

1. Ansible Ad-Hoc Commands:
We can use command module or SELinux module in Ansible to manage the SELinux tasks on the Ansible managed nodes.
Using Command Module:
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m command -a 'getenforce' -b
$ ansible node1 -m command -a 'setenforce 0' -b
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m command -a 'getenforce' -b
$ ansible node1 -m command -a 'setenforce 1' -b
$ ansible node1 -m command -a 'getenforce' -b
$ ansible node1 -m command -a 'ls -lZ /var/log/messages' -b
$ ansible node1 -m command -a 'chcon -t httpd_sys_content_t /var/log/messages' -b

Manage SELinux Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
$ ansible node1 -m command -a 'ls -lZ /var/log/messages' -b
$ ansible node1 -m command -a 'restorecon -v /var/log/messages' -b
$ ansible node1 -m command -a 'ls -lZ /var/log/messages' -b
$ ansible node1 -m command -a 'ls -lZ /etc/yum.repos.d/local.repo' -b
$ ansible node1 -m command -a 'touch /tmp/file.txt' -b
$ ansible node1 -m command -a 'ls -lZ /tmp/file.txt' -b
$ ansible node1 -m command -a 'chcon -t var_log_t /tmp/file.txt' -b
$ ansible node1 -m command -a 'ls -lZ /tmp/file.txt' -b
$ ansible node1 -m command -a 'id' -b
$ ansible node1 -m command -a 'ps -eflZ' -b | grep bash
$ ansible node1 -m command -a 'getsebool -a' -b | grep ssh
$ ansible node1 -m command -a 'setsebool -P ssh_keysign on' -b
$ ansible node1 -m command -a 'getsebool ssh_keysign' -b
$ ansible node1 -m command -a 'semanage port -l' -b
$ ansible node1 -m command -a 'semanage port -l' -b | grep ssh_port_t
```

Using SELinux Module:

```
$ ansible-galaxy collection install ansible.posix
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m selinux -a 'policy=targeted state=permissive' -b
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m selinux -a 'policy=targeted state=enforcing' -b
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m selinux -a 'state=disabled' -b
$ ansible node1 -m command -a 'sestatus' -b
$ ansible node1 -m command -a 'ls -lZ /tmp/file.txt' -b
```

Using Sefcontext Module:

```
$ ansible-galaxy collection install community.general
$ ansible node1 -m sefcontext -a 'target=/tmp/file.txt setype=httpd_sys_rw_content_t state=present' -b
$ ansible node1 -m command -a 'ls -lZ /tmp/file.txt' -b
$ ansible node1 -m command -a 'semanage fcontext -l' -b | grep /tmp/file.txt
$ ansible node1 -m command -a 'restorecon -v /tmp/file.txt' -b
$ ansible node1 -m command -a 'ls -lZ /tmp/file.txt' -b
$ ansible node1 -m sefcontext -a 'target=/tmp/file.txt setype=httpd_sys_rw_content_t state=absent' -b
$ ansible node1 -m command -a 'semanage fcontext -l' -b | grep /tmp/file.txt
```

Using Seboolean Module:

```
$ ansible node1 -m command -a 'getsebool -a' -b | grep httpd_can_connect_ftp
$ ansible node1 -m command -a 'getsebool httpd_can_connect_ftp' -b
$ ansible node1 -m seboolean -a 'name=httpd_can_connect_ftp state=yes persistent=yes' -b
$ ansible node1 -m command -a 'getsebool httpd_can_connect_ftp' -b
```

Using Seport Module:

```
$ ansible node1 -m command -a 'semanage port -l' -b | grep ssh
$ ansible node1 -m seport -a 'ports=20222 proto=tcp setype=ssh_port_t state=present' -b
$ ansible node1 -m command -a 'semanage port -l' -b | grep ssh
```

Manage SELinux Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

2. Ansible Playbooks:

We can use ansible playbooks to manage the SELinux tasks on the managed nodes.

```
$ vim selinux.yml
---
- name: Manage SELinux
  hosts: node1
  become: true
  tasks:
    - name: Set SELinux to Permissive mode
      ansible.posix.selinux:
        policy: targeted
        state: permissive
...

$ ansible node1 -m command -a 'sestatus' -b
$ ansible-playbook selinux.yml
$ ansible node1 -m command -a 'sestatus' -b

$ vim selinux2.yml
---
- name: Manage SELinux
  hosts: node1
  become: true
  tasks:
    - name: Set SELinux to Permissive mode
      ansible.posix.selinux:
        policy: targeted
        state: enforcing
...

$ ansible-playbook selinux2.yml
$ ansible node1 -m command -a 'sestatus' -b

$ vim selinux3.yml
---
- name: Manage SELinux
  hosts: node1
  become: true
  tasks:
    - name: Allow Apache to modify files in /tmp directory
      community.general.sefcontext:
        target: '/tmp(/.*)?'
        setype: httpd_sys_rw_content_t
        state: present
    - name: Apply SELinux file context
      ansible.builtin.command: restorecon -irv /tmp
...

$ ansible-playbook selinux3.yml
$ ansible node1 -m command -a 'ls -lZ /tmp' -b
```

Manage SELinux Using Ansible

By: Er. Vikas Nehra (M. Tech, B. Tech), Experience: 15 + Years

```
$ vim selinux4.yml
---
- name: Manage SELinux
  hosts: node1
  become: true
  tasks:
    - name: Allow sshd to listen on tcp port 20224
      community.general.seport:
        ports: 20224
        proto: tcp
        setype: ssh_port_t
        state: present

    - name: Allow memcached to listen on a range of ports
      community.general.seport:
        ports: 10000-10100,10112
        proto: tcp
        setype: memcache_port_t
        state: present
...

$ ansible node1 -m command -a 'semanage port -l' -b | grep ssh
$ ansible node1 -m command -a 'semanage port -l' -b | grep memcache
$ ansible-playbook selinux4.yml
$ ansible node1 -m command -a 'semanage port -l' -b | grep ssh
$ ansible node1 -m command -a 'semanage port -l' -b | grep memcache

$ vim selinux5.yml
---
- name: Manage SELinux
  hosts: node1
  become: true
  tasks:
    - name: Set httpd_can_connect_ftp flag ON and enable it to persist on reboot
      ansible.posix.seboolean:
        name: httpd_can_connect_ftp
        state: yes
        persistent: yes
...

$ ansible node1 -m command -a 'getsebool -a' -b | grep httpd_can_connect_ftp
$ ansible-playbook selinux5.yml
$ ansible node1 -m command -a 'getsebool -a' -b | grep httpd_can_connect_ftp
```

Thank You