OTUS

Методическое пособие
по выполнению домашнего задания курса
«Администратор Linux. Professional»

Стенд с Vagrant с SELinux

Содержание

1. Введение

SElinux (Security Enhanced Linux) – система принудительного (мандатного) контроля доступа (MAC). Разрабатывалась АНБ. В 2003 году вошла в состав ядра linux 2.6.x.
SELinux следует модели минимально необходимых привилегий для каждого сервиса пользователя и программы.

Зачем нужен Selinux:

- Гибкое ограничение прав пользователей и процессов на уровне ядра
- Работа совместно с DAC (матричным управлением доступа)
- Снижение риска, возникающего вследствие допущенных ошибок
- Ограничение потенциально опасных или скомпрометированных процессов в правах
- Протоколирование

Обычно в В Selinux большие и сложные политики. Каждый ресурс должен быть описан и сопоставлен с сервисом.

Режимы работы SELinux:
Enforcing (по-умолчанию) – Активная работа. Всё, что нарушает политику безопасности блокируется. Попытка нарушения фиксируется в журнале.
Permissive — запрещенные действия не блокируются. Все нарушения пишутся в журнал
Disabled — полное отключение SELinux.

Важно помнить: классическая система прав Unix применяется первой и управление перейдёт к SELinux только в том случае, если эта первичная проверка будет успешно пройдена.

Написать политику SELinux достаточно сложно, но для ключевых приложений и сервисов (например httpd, mysqld, dhcpd и т. д.) определены заранее сконфигурированные политики, которые не позволят получить злоумышленнику доступ к важным данным.
Те приложения, для которых политика не определена, выполняются в домене unconfined_f и не защищаются SELinux.

2. Цели домашнего задания

Диагностировать проблемы и модифицировать политики SELinux для корректной работы приложений, если это требуется.

3. Описание домашнего задания

1. Запустить nginx на нестандартном порту 3-мя разными способами:
переключатели setsebool;
добавление нестандартного порта в имеющийся тип;
формирование и установка модуля SELinux.

К сдаче:
README с описанием каждого решения
(скриншоты и демонстрация приветствуются).

2. Обеспечить работоспособность приложения при включенном selinux.
развернуть приложенный стенд
https://github.com/mbfx/otus-linux-adm/tree/master/selinux_dns_problems;
выяснить причину неработоспособности механизма обновления зоны (см. README);
предложить решение (или решения) для данной проблемы;
выбрать одно из решений для реализации, предварительно обосновав выбор;
реализовать выбранное решение и продемонстрировать его работоспособность.

4. Пошаговая инструкция выполнения домашнего задания

ПК на Unix с 8ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.

Предварительно установленное и настроенное следующее ПО:

Hashicorp Vagrant (https://www.vagrantup.com/downloads)

Oracle VirtualBox (https://www.virtualbox.org/wiki/Linux_Downloads).

Ansible (версия 2.7 и выше) - https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Git - https://git-scm.com/book/ru/v2/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-Git

1. Создаём виртуальную машину

Создаём каталог, в котором будут храниться настройки виртуальной машины. В каталоге создаём файл с именем Vagrantfile, добавляем в него следующее содержимое:

```
# -*- mode: ruby -*-
# vim: set ft=ruby :

MACHINES = {
  :selinux => {
        :box_name => "centos/7",
        :box_version => "2004.01",
        #:provision => "test.sh",
  },
}

Vagrant.configure("2") do |config|

  MACHINES.each do |boxname, boxconfig|

      config.vm.define boxname do |box|

        box.vm.box = boxconfig[:box_name]
        box.vm.box_version = boxconfig[:box_version]

        box.vm.host_name = "selinux"
        box.vm.network "forwarded_port", guest: 4881, host: 4881

        box.vm.provider :virtualbox do |vb|
              vb.customize ["modifyvm", :id, "--memory", "1024"]
              needsController = false
        end
```

```
        box.vm.provision "shell", inline: <<-SHELL
          #install epel-release
          yum install -y epel-release
          #install nginx
          yum install -y nginx
          #change nginx port
          sed -ie 's/:80/:4881/g' /etc/nginx/nginx.conf
          sed -i 's/listen       80;/listen       4881;/'
/etc/nginx/nginx.conf
          #disable SELinux
          #setenforce 0
          #start nginx
          systemctl start nginx
          systemctl status nginx
          #check nginx port
          ss -tlpn | grep 4881
        SHELL
    end
  end
end
```

Результатом выполнения команды *vagrant up* станет созданная виртуальная машина с установленным nginx, который работает на порту TCP 4881. Порт TCP 4881 уже проброшен до хоста. SELinux включен.

Во время развёртывания стенда попытка запустить nginx завершится с ошибкой:

```
selinux: ● nginx.service - The nginx HTTP and reverse proxy server
    selinux:    Loaded: loaded (/usr/lib/systemd/system/nginx.service;
disabled; vendor preset: disabled)
    selinux:    Active: failed (Result: exit-code) since Sun 2021-11-07
02:19:25 UTC; 10ms ago
    selinux:   Process: 2811 ExecStartPre=/usr/sbin/nginx -t
(code=exited, status=1/FAILURE)
    selinux:   Process: 2810 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
(code=exited, status=0/SUCCESS)
    selinux:
    selinux: Nov 07 02:19:25 selinux systemd[1]: Starting The nginx
HTTP and reverse proxy server...
    selinux: Nov 07 02:19:25 selinux nginx[2811]: nginx: the
configuration file /etc/nginx/nginx.conf syntax is ok
    selinux: Nov 07 02:19:25 selinux nginx[2811]: nginx: [emerg] bind()
to 0.0.0.0:4881 failed (13: Permission denied)
    selinux: Nov 07 02:19:25 selinux nginx[2811]: nginx: configuration
file /etc/nginx/nginx.conf test failed
    selinux: Nov 07 02:19:25 selinux systemd[1]: nginx.service: control
process exited, code=exited status=1
    selinux: Nov 07 02:19:25 selinux systemd[1]: Failed to start The
nginx HTTP and reverse proxy server.
    selinux: Nov 07 02:19:25 selinux systemd[1]: Unit nginx.service
entered failed state.
    selinux: Nov 07 02:19:25 selinux systemd[1]: nginx.service failed.
```

Данная ошибка появляется из-за того, что SELinux блокирует работу nginx на нестандартном порту.

Заходим на сервер: *vagrant ssh*

Дальнейшие действия выполняются от пользователя root. Переходим в root пользователя: *sudo -i*

2. **Запуск nginx на нестандартном порту 3-мя разными способами**

Для начала проверим, что в ОС отключен файервол: *systemctl status firewalld*

```
[root@selinux ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled;
vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
[root@selinux ~]#
```

Также можно проверить, что конфигурация nginx настроена без ошибок: *nginx -t*

```
[root@selinux ~]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[root@selinux ~]#
```

Далее проверим режим работы SELinux: getenforce

```
[root@selinux ~]# getenforce
 Enforcing
[root@selinux ~]#
```

Должен отображаться режим Enforcing. Данный режим означает, что SELinux будет блокировать запрещенную активность.

Разрешим в SELinux работу nginx на порту TCP 4881 с помощью переключателей setsebool

Находим в логах (/var/log/audit/audit.log) информацию о блокировании порта

```
type=AVC msg=audit(1636489992.273:967): avc:  denied  { name_bind } for  pid=22278 comm="nginx" src=4881 scontext=system_u:system_r:h
ttpd_t:s0 tcontext=system_u:object_r:unreserved_port_t:s0 tclass=tcp_socket permissive=0
type=SYSCALL msg=audit(1636489992.273:967): arch=c000003e syscall=49 success=no exit=-13 a0=6 a1=55fac67217b8 a2=10 a3=7fff265ac9a0 i
tems=0 ppid=1 pid=22278 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="nginx
" exe="/usr/sbin/nginx" subj=system_u:system_r:httpd_t:s0 key=(null)
type=PROCTITLE msg=audit(1636489992.273:967): proctitle=2F7573722F7362696E2F6E67696E78002D74
type=SERVICE_START msg=audit(1636489992.277:968): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='un
it=nginx comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=failed'
(END)
```

Копируем время, в которое был записан этот лог, и, с помощью утилиты audit2why смотрим информации о запрете: *grep 1636489992.273:967 /var/log/audit/audit.log | audit2why*

```
[root@selinux ~]# grep 1636489992.273:967 /var/log/audit/audit.log |
audit2why
type=AVC msg=audit(1636489992.273:967): avc:  denied  { name_bind } for
pid=22278 comm="nginx" src=4881 scontext=system_u:system_r:httpd_t:s0
```

```
tcontext=system_u:object_r:unreserved_port_t:s0 tclass=tcp_socket
permissive=0

        Was caused by:
        The boolean nis_enabled was set incorrectly.
        Description:
        Allow nis to enabled

        Allow access by executing:
        # setsebool -P nis_enabled 1
[root@selinux ~]#
```

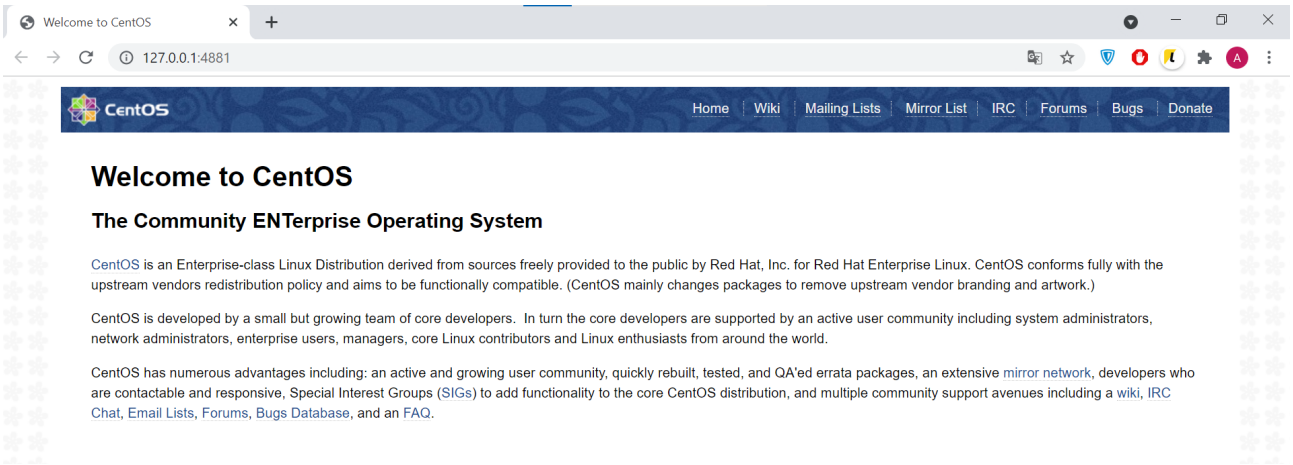
Утилита audit2why покажет почему трафик блокируется. Исходя из вывода утилиты, мы видим, что нам нужно поменять параметр nis_enabled.

Включим параметр nis_enabled и перезапустим nginx: *setsebool -P nis_enabled on*

```
[root@selinux ~]# setsebool -P nis_enabled on
[root@selinux ~]# systemctl restart nginx
[root@selinux ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
vendor preset: disabled)
   Active: active (running) since Tue 2021-11-09 20:45:41 UTC; 6s ago
  Process: 22327 ExecStart=/usr/sbin/nginx (code=exited,
status=0/SUCCESS)
  Process: 22324 ExecStartPre=/usr/sbin/nginx -t (code=exited,
status=0/SUCCESS)
  Process: 22323 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
(code=exited, status=0/SUCCESS)
 Main PID: 22329 (nginx)
   CGroup: /system.slice/nginx.service
           ├─22329 nginx: master process /usr/sbin/nginx
           └─22331 nginx: worker process

Nov 09 20:45:41 selinux systemd[1]: Starting The nginx HTTP and reverse
proxy server...
Nov 09 20:45:41 selinux nginx[22324]: nginx: the configuration file
/etc/nginx/nginx.conf syntax is ok
Nov 09 20:45:41 selinux nginx[22324]: nginx: configuration file
/etc/nginx/nginx.conf test is successful
Nov 09 20:45:41 selinux systemd[1]: Started The nginx HTTP and reverse
proxy server.
[root@selinux ~]#
```

Также можно проверить работу nginx из браузера. Заходим в любой браузер на хосте и переходим по адресу http://127.0.0.1:4881



Проверить статус параметра можно с помощью команды: *getsebool -a | grep nis_enabled*

```
[root@selinux ~]# getsebool -a | grep nis_enabled
nis_enabled --> on
[root@selinux ~]#
```

Вернём запрет работы nginx на порту 4881 обратно. Для этого отключим nis_enabled: *setsebool -P nis_enabled off*

После отключения nis_enabled служба nginx снова не запустится.

Теперь разрешим в SELinux работу nginx на порту TCP 4881 с помощью добавления нестандартного порта в имеющийся тип:

Поиск имеющегося типа, для http трафика: *semanage port -l | grep http*

```
[root@selinux ~]# semanage port -l | grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009,
8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
[root@selinux ~]#
```

Добавим порт в тип http_port_t: *semanage port -a -t http_port_t -p tcp 4881*

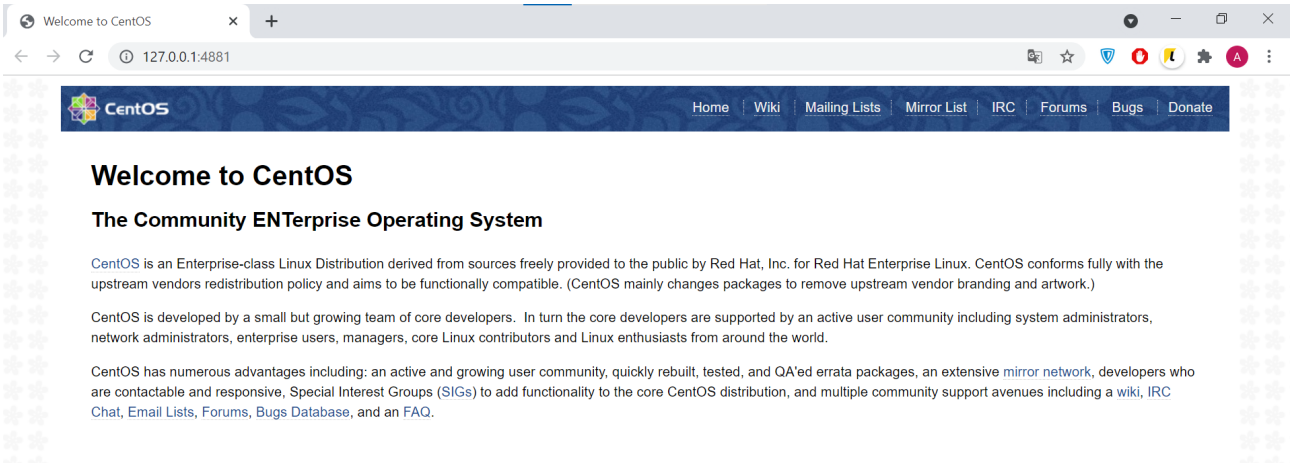
```
[root@selinux ~]# semanage port -a -t http_port_t -p tcp 4881
[root@selinux ~]# semanage port -l | grep  http_port_t
http_port_t                    tcp      4881, 80, 81, 443, 488, 8008,
8009, 8443, 9000
pegasus_http_port_t            tcp      5988
[root@selinux ~]#
```

Теперь перезапустим службу nginx и проверим её работу: *systemctl restart nginx*

```
[root@selinux ~]# systemctl restart nginx
[root@selinux ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
vendor preset: disabled)
   Active: active (running) since Sun 2021-11-07 02:52:59 UTC; 5s ago
  Process: 2981 ExecStart=/usr/sbin/nginx (code=exited,
status=0/SUCCESS)
  Process: 2979 ExecStartPre=/usr/sbin/nginx -t (code=exited,
status=0/SUCCESS)
  Process: 2978 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
(code=exited, status=0/SUCCESS)
 Main PID: 2983 (nginx)
   CGroup: /system.slice/nginx.service
           ├─2983 nginx: master process /usr/sbin/nginx
           └─2985 nginx: worker process

Nov 07 02:52:59 selinux systemd[1]: Starting The nginx HTTP and reverse
proxy server...
Nov 07 02:52:59 selinux nginx[2979]: nginx: the configuration file
/etc/nginx/nginx.conf syntax is ok
Nov 07 02:52:59 selinux nginx[2979]: nginx: configuration file
/etc/nginx/nginx.conf test is successful
Nov 07 02:52:59 selinux systemd[1]: Started The nginx HTTP and reverse
proxy server.
[root@selinux ~]#
```

Также можно проверить работу nginx из браузера. Заходим в любой браузер на хосте и переходим по адресу http://127.0.0.1:4881



Удалить нестандартный порт из имеющегося типа можно с помощью команды: *semanage port -d -t http_port_t -p tcp 4881*

```
[root@selinux ~]# semanage port -d -t http_port_t -p tcp 4881
[root@selinux ~]# semanage port -l | grep  http_port_t
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009,
8443, 9000
pegasus_http_port_t            tcp      5988
[root@selinux ~]# systemctl restart nginx
```

```
Job for nginx.service failed because the control process exited with
error code. See "systemctl status nginx.service" and "journalctl -xe"
for details.

[root@selinux ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
vendor preset: disabled)
   Active: failed (Result: exit-code) since Sun 2021-11-07 03:00:42
UTC; 3s ago
...
Nov 07 03:00:42 selinux nginx[3008]: nginx: the configuration file
/etc/nginx/nginx.conf syntax is ok
Nov 07 03:00:42 selinux nginx[3008]: nginx: [emerg] bind() to
0.0.0.0:4881 failed (13: Permission denied)
...
Nov 07 03:00:42 selinux systemd[1]: nginx.service failed.
[root@selinux ~]#
```

Разрешим в SELinux работу nginx на порту TCP 4881 с помощью формирования и установки модуля SELinux:

Попробуем снова запустить nginx: *systemctl start nginx*

```
[root@selinux ~]# systemctl start nginx
Job for nginx.service failed because the control process exited with
error code. See "systemctl status nginx.service" and "journalctl -xe"
for details.
[root@selinux ~]#
```

Nginx не запуститься, так как SELinux продолжает его блокировать. Посмотрим логи SELinux, которые относятся к nginx:

```
[root@selinux ~]# grep nginx /var/log/audit/audit.log
...
type=SYSCALL msg=audit(1637045467.417:510): arch=c000003e syscall=49
success=no exit=-13 a0=6 a1=558922a5a7b8 a2=10 a3=7ffe62da3900 items=0
ppid=1 pid=2133 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0
egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="nginx"
exe="/usr/sbin/nginx" subj=system_u:system_r:httpd_t:s0 key=(null)
type=SERVICE_START msg=audit(1637045467.419:511): pid=1 uid=0
auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0
msg='unit=nginx comm="systemd" exe="/usr/lib/systemd/systemd"
hostname=? addr=? terminal=? res=failed'
[root@selinux ~]#
```

Воспользуемся утилитой audit2allow для того, чтобы на основе логов SELinux сделать модуль, разрешающий работу nginx на нестандартном порту:

grep nginx /var/log/audit/audit.log | audit2allow -M nginx

```
[root@selinux ~]# grep nginx /var/log/audit/audit.log | audit2allow -M
nginx
******************** IMPORTANT ************************
```

```
To make this policy package active, execute:

semodule -i nginx.pp

[root@selinux ~]#
```

Audit2allow сформировал модуль, и сообщил нам команду, с помощью которой можно применить данный модуль: *semodule -i nginx.pp*

```
[root@selinux ~]# semodule -i nginx.pp
[root@selinux ~]#
```

Попробуем снова запустить nginx: *systemctl start nginx*

```
[root@selinux ~]# systemctl start nginx
[root@selinux ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled;
vendor preset: disabled)
   Active: active (running) since Tue 2021-11-16 06:59:56 UTC; 16s ago
  Process: 2163 ExecStart=/usr/sbin/nginx (code=exited,
status=0/SUCCESS)
  Process: 2161 ExecStartPre=/usr/sbin/nginx -t (code=exited,
status=0/SUCCESS)
  Process: 2160 ExecStartPre=/usr/bin/rm -f /run/nginx.pid
(code=exited, status=0/SUCCESS)
 Main PID: 2165 (nginx)
   CGroup: /system.slice/nginx.service
           ├─2165 nginx: master process /usr/sbin/nginx
           └─2167 nginx: worker process

Nov 16 06:59:55 selinux systemd[1]: Starting The nginx HTTP and reverse
proxy server...
Nov 16 06:59:56 selinux nginx[2161]: nginx: the configuration file
/etc/nginx/nginx.conf syntax is ok
Nov 16 06:59:56 selinux nginx[2161]: nginx: configuration file
/etc/nginx/nginx.conf test is successful
Nov 16 06:59:56 selinux systemd[1]: Started The nginx HTTP and reverse
proxy server.
[root@selinux ~]#
```

После добавления модуля nginx запустился без ошибок. При использовании модуля изменения сохранятся после перезагрузки.

Просмотр всех установленных модулей: *semodule -l*

Для удаления модуля воспользуемся командой: *semodule -r nginx*

```
[root@selinux ~]# semodule -r nginx
libsemanage.semanage_direct_remove_key: Removing last nginx module (no
other nginx module exists at another priority).
[root@selinux ~]#
```

Результатом выполнения данного задания будет подготовленная документация.

Документация

Создайте файл README.md и снабдите его следующей информацией:
- название выполняемого задания;
- текст задания;
- полное описание всех команд;
- скриншоты (если потребуется);
- заметки, если считаете, что имеет смысл их зафиксировать в репозитории.

3. Обеспечение работоспособности приложения при включенном SELinux

Для того, чтобы развернуть стенд потребуется хост, с установленным git и ansible.

Инструкция по установке Ansible - https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

Инструкция по установке Git - https://git-scm.com/book/ru/v2/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5-%D0%A3%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-Git

Выполним клонирование репозитория: *git clone https://github.com/mbfx/otus-linux-adm.git*

```
➜ ~ git clone https://github.com/mbfx/otus-linux-adm.git
Cloning into 'otus-linux-adm'...
remote: Enumerating objects: 542, done.
remote: Counting objects: 100% (440/440), done.
remote: Compressing objects: 100% (295/295), done.
remote: Total 542 (delta 118), reused 381 (delta 69), pack-reused 102
Receiving objects: 100% (542/542), 1.38 MiB | 3.65 MiB/s, done.
Resolving deltas: 100% (133/133), done.
```

Перейдём в каталог со стендом: *cd otus-linux-adm/selinux_dns_problems*

Развернём 2 ВМ с помощью vagrant: *vagrant up*

После того, как стенд развернется, проверим ВМ с помощью команды: *vagrant status*

```
➜ selinux_dns_problems (master) ✔ vagrant status
Current machine states:

ns01                      running (virtualbox)
client                    running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
➜ selinux_dns_problems (master) ✔
```

Подключимся к клиенту: *vagrant ssh client*

Попробуем внести изменения в зону: *nsupdate -k /etc/named.zonetransfer.key*

```
[vagrant@client ~]$ nsupdate -k /etc/named.zonetransfer.key
> server 192.168.50.10
> zone ddns.lab
> update add www.ddns.lab. 60 A 192.168.50.15
> send
update failed: SERVFAIL
> quit
[vagrant@client ~]$
```

Изменения внести не получилось. Давайте посмотрим логи SELinux, чтобы понять в чём может быть проблема.

Для этого воспользуемся утилитой audit2why: *cat /var/log/audit/audit.log | audit2why*

```
[vagrant@client ~]$ sudo -i
[root@client ~]# cat /var/log/audit/audit.log | audit2why
[root@client ~]#
```

Тут мы видим, что на клиенте отсутствуют ошибки.

Не закрывая сессию на клиенте, подключимся к серверу ns01 и проверим логи SELinux:

```
➜ selinux_dns_problems (master) ✔ vagrant ssh ns01
Last login: Tue Nov 16 09:58:37 2021 from 10.0.2.2
[vagrant@ns01 ~]$ sudo -i
[root@ns01 ~]#
[root@ns01 ~]#
[root@ns01 ~]# cat /var/log/audit/audit.log | audit2why
type=AVC msg=audit(1637070345.890:1972): avc:  denied  { create } for
pid=5192 comm="isc-worker0000" name="named.ddns.lab.view1.jnl"
scontext=system_u:system_r:named_t:s0 tcontext=system_u:object_r:etc_t:s0
tclass=file permissive=0

    Was caused by:
        Missing type enforcement (TE) allow rule.

        You can use audit2allow to generate a loadable module to allow
this access.

[root@ns01 ~]#
```

В логах мы видим, что ошибка в контексте безопасности. Вместо типа **named_t** используется тип **etc_t**. Проверим данную проблему в каталоге /etc/named

```
[root@ns01 ~]# ls -laZ /etc/named
drw-rwx---. root named system_u:object_r:etc_t:s0       .
drwxr-xr-x. root root  system_u:object_r:etc_t:s0       ..
drw-rwx---. root named unconfined_u:object_r:etc_t:s0   dynamic
-rw-rw----. root named system_u:object_r:etc_t:s0
named.50.168.192.rev
-rw-rw----. root named system_u:object_r:etc_t:s0       named.dns.lab
-rw-rw----. root named system_u:object_r:etc_t:s0
named.dns.lab.view1
-rw-rw----. root named system_u:object_r:etc_t:s0
named.newdns.lab
```

```
[root@ns01 ~]#
```

Тут мы также видим, что контекст безопасности неправильный. Проблема заключается в том, что конфигурационные файлы лежат в другом каталоге. Посмотреть в каком каталоги должны лежать, файлы, чтобы на них распространялись правильные политики SELinux можно с помощью команды: *sudo semanage fcontext -l | grep named*

```
[root@ns01 ~]# sudo semanage fcontext -l | grep named
/etc/rndc.*               regular file
system_u:object_r:named_conf_t:s0
/var/named(/.*)?          all files
system_u:object_r:named_zone_t:s0
...
[root@ns01 ~]#
```

Изменим тип контекста безопасности для каталога /etc/named: *sudo chcon -R -t named_zone_t /etc/named*

```
[root@ns01 ~]# sudo chcon -R -t named_zone_t /etc/named
[root@ns01 ~]#
[root@ns01 ~]# ls -laZ /etc/named
drw-rwx---. root named system_u:object_r:named_zone_t:s0 .
drwxr-xr-x. root root  system_u:object_r:etc_t:s0       ..
drw-rwx---. root named unconfined_u:object_r:named_zone_t:s0 dynamic
-rw-rw----. root named system_u:object_r:named_zone_t:s0
named.50.168.192.rev
-rw-rw----. root named system_u:object_r:named_zone_t:s0 named.dns.lab
-rw-rw----. root named system_u:object_r:named_zone_t:s0
named.dns.lab.view1
-rw-rw----. root named system_u:object_r:named_zone_t:s0
named.newdns.lab
[root@ns01 ~]#
```

Попробуем снова внести изменения с клиента:

```
[vagrant@client ~]$ nsupdate -k /etc/named.zonetransfer.key
> server 192.168.50.10
> zone ddns.lab
> update add www.ddns.lab. 60 A 192.168.50.15
> send
> quit
[vagrant@client ~]$
[vagrant@client ~]$ dig www.ddns.lab

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.7 <<>> www.ddns.lab
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52762
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.ddns.lab.          IN   A
```

```
;; ANSWER SECTION:
www.ddns.lab.        60   IN   A    192.168.50.15

;; AUTHORITY SECTION:
ddns.lab.        3600     IN   NS   ns01.dns.lab.

;; ADDITIONAL SECTION:
ns01.dns.lab.        3600     IN   A    192.168.50.10

;; Query time: 1 msec
;; SERVER: 192.168.50.10#53(192.168.50.10)
;; WHEN: Thu Nov 18 10:34:41 UTC 2021
;; MSG SIZE  rcvd: 96

[vagrant@client ~]$
```

Видим, что изменения применились. Попробуем перезагрузить хосты и ещё раз сделать запрос с помощью dig:

```
[vagrant@client ~]$ dig @192.168.50.10 www.ddns.lab

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el7_9.7 <<>> @192.168.50.10
www.ddns.lab
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52392
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.ddns.lab.          IN   A

;; ANSWER SECTION:
www.ddns.lab.        60   IN   A    192.168.50.15

;; AUTHORITY SECTION:
ddns.lab.        3600     IN   NS   ns01.dns.lab.

;; ADDITIONAL SECTION:
ns01.dns.lab.        3600     IN   A    192.168.50.10

;; Query time: 2 msec
;; SERVER: 192.168.50.10#53(192.168.50.10)
;; WHEN: Thu Nov 18 15:49:07 UTC 2021
;; MSG SIZE  rcvd: 96

[vagrant@client ~]$
```

Всё правильно. После перезагрузки настройки сохранились.

Для того, чтобы вернуть правила обратно, можно ввести команду:
restorecon -v -R /etc/named

```
[root@ns01 ~]# restorecon -v -R /etc/named
```

```
restorecon reset /etc/named context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/named.dns.lab.view1 context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/named.dns.lab context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/dynamic context
unconfined_u:object_r:named_zone_t:s0->unconfined_u:object_r:etc_t:s0
restorecon reset /etc/named/dynamic/named.ddns.lab context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/dynamic/named.ddns.lab.view1 context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/dynamic/named.ddns.lab.view1.jnl context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/named.newdns.lab context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
restorecon reset /etc/named/named.50.168.192.rev context
system_u:object_r:named_zone_t:s0->system_u:object_r:etc_t:s0
[root@ns01 ~]#
```

Результатом выполнения данного задания будет:

- README с анализом причины неработоспособности, возможными способами решения и обоснованием выбора одного из них;
- исправленный стенд или демонстрация работоспособной системы скриншотами и описанием.

5. Критерий оценивания

Статус "Принято" ставится при выполнении следующих условий:

1. Ссылка на репозиторий GitHub.
2. Vagrantfile с шагами установки необходимых компонентов
3. Исходный код scripts для настройки сервера (если необходимо)
4. По заданию 1 подготовлена документация в которой реализованы и продемонстрированы все 3 способа решения.
5. Для задания 2 подготовлена документация, в которой описана причина неработоспособности механизма обновления зоны и продемонстрирован один из способов решения.

Опционально для выполнения:
- Для задания 2 предложено более одного способа решения
- для задания 2 обосновать один из способов решения

6. Рекомендуемые источники

Видео «Использование SELinux»
https://www.youtube.com/watch?v=EKCe7-6VrOY&t=350s&ab_channel=OTUS%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD-%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

Статья «Руководство для начинающих по SELinux»
https://habr.com/ru/company/otus/blog/460387/

Статья «SELinux – описание и особенности работы с системой. Часть 1» - https://habr.com/ru/company/kingservers/blog/209644/
Статья «SELinux — описание и особенности работы с системой. Часть 2» - https://habr.com/ru/company/kingservers/blog/209970/
SELinux Wiki - http://www.selinuxproject.org/page/Main_Page