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

Vagrant-стенд с PAM

Цель домашнего задания

Научиться создавать пользователей и добавлять им ограничения

Описание домашнего задания

Ограничить доступ к системе для всех пользователей, кроме группы администраторов, в выходные дни (суббота и воскресенье), за исключением праздничных дней.

⭐ **Задание со звездочкой**
Предоставить определённому пользователю доступ к Docker и право перезапускать Docker-сервис.

Введение

Почти все операционные системы Linux — многопользовательские. Администратор Linux должен уметь создать и настраивать пользователей.

В Linux есть 3 группы пользователей:

- Администраторы — привилегированные пользователи с полным доступом к системе. По умолчанию в ОС есть такой пользователь — root
- Локальные пользователи — их учетные записи создает администратор, их права ограничены. Администраторы могут изменять права локальных пользователей
- Системные пользователи — учетный записи, которые создаются системой для внутренних процессов и служб. Например пользователь — nginx

У каждого пользователя есть свой уникальный идентификатор — UID.

Чтобы упростить процесс настройки прав для новых пользователей, их объединяют в группы. Каждая группа имеет свой набор прав и ограничений. Любой пользователь, создаваемый или добавляемый в такую группу, автоматически их наследует. Если при добавлении пользователя для него не указать группу, то у него будет своя, индивидуальная группа — с именем пользователя. Один пользователь может одновременно входить в несколько групп.

Информацию о каждом пользователе сервера можно посмотреть в файле /etc/passwd

Для более точных настроек пользователей можно использовать подключаемые модули аутентификации (PAM)

PAM (Pluggable Authentication Modules - подключаемые модули аутентификации) — набор библиотек, которые позволяют интегрировать различные методы аутентификации в виде единого API.

PAM решает следующие задачи:

- Аутентификация — процесс подтверждения пользователем своей подлинности. Например: ввод логина и пароля, ssh-ключ и т д.
- Авторизация — процесс наделения пользователя правами
- Отчетность — запись информации о произошедших событиях

PAM может быть реализован несколькими способами:

- Модуль pam_time — настройка доступа для пользователя с учетом времени
- Модуль pam_exec — настройка доступа для пользователей с помощью скриптов

- И т.д.

Формат сдачи:

Vagrantfile + ansible

Критерии оценивания

Статус «Принято» ставится при выполнении следующих условий:

1. Ссылка на репозиторий GitHub.
2. Vagrantfile, который будет разворачивать виртуальную машину
3. Документация по каждому заданию:
Создайте файл README.md и снабдите его следующей информацией:
- название выполняемого задания;
- текст задания;
- описание команд и их вывод;
- особенности проектирования и реализации решения,
- заметки, если считаете, что имеет смысл их зафиксировать в репозитории.

Функциональные и нефункциональные требования

- ПК на Unix с 8ГБ ОЗУ или виртуальная машина с включенной Nested Virtualization.
- Созданный аккаунт на GitHub - https://github.com/
- Если Вы находитесь в России, для корректной работы Вам может потребоваться VPN.

Предварительно установленное и настроенное следующее ПО:

- Hashicorp Vagrant (https://www.vagrantup.com/downloads)
- Oracle VirtualBox (https://www.virtualbox.org/wiki/Linux_Downloads).
- Любой редактор кода, например Visual Studio Code, Atom и т.д.

Инструкция по выполнению домашнего задания

Все дальнейшие действия были проверены при использовании Vagrant 2.2.19, VirtualBox v6.1.26 r145957. В качестве ОС на хостах установлена Ubuntu 22.04. Серьёзные отступления от этой конфигурации могут потребовать адаптации с вашей стороны.

Создадим Vagrantfile, в котором будут указаны параметры наших ВМ:

```
MACHINES = {
  :"pam" => {
            :box_name => "ubuntu/jammy64",
            :cpus => 2,
            :memory => 1024,
            :ip => "192.168.57.10",
           }
}

Vagrant.configure("2") do |config|
  MACHINES.each do |boxname, boxconfig|
    config.vm.synced_folder ".", "/vagrant", disabled: true
    config.vm.network "private_network", ip: boxconfig[:ip]
    config.vm.define boxname do |box|
```

```
    box.vm.box = boxconfig[:box_name]
    box.vm.box_version = boxconfig[:box_version]
    box.vm.host_name = boxname.to_s

    box.vm.provider "virtualbox" do |v|
      v.memory = boxconfig[:memory]
      v.cpus = boxconfig[:cpus]
    end
    box.vm.provision "shell", inline: <<-SHELL
       sed -i 's/^PasswordAuthentication.*$/PasswordAuthentication yes/' /etc/ssh/sshd_config
       systemctl restart sshd.service
          SHELL
  end
 end
end
```

После создания Vagrantfile запустим нашу ВМ командой *vagrant up*. Будет создана одна виртуальная машина.

Уточним некоторые моменты из Vagrantfile:

- Так как в данной лабораторной работе нам предстоит подключаться к нашей ВМ в неё добавлен дополнительный сетевой интерфейс:

```
...
        # Указываем IP-адрес для ВМ
        :ip => "192.168.57.10",
...
 # Добавляем сетевой интерфейс
  config.vm.network "private_network", ip: boxconfig[:ip]
...
```

- Для удобства, в параметрах SSH разрешена аутентификация пользователя по паролю:

```
box.vm.provision "shell", inline: <<-SHELL
      #Разрешаем подключение пользователей по SSH с использованием пароля
      sed -i 's/^PasswordAuthentication.*$/PasswordAuthentication yes/' /etc/ssh/sshd_config
      #Перезапуск службы SSHD
      systemctl restart sshd.service
         SHELL
```

Настройка запрета для всех пользователей (кроме группы Admin) логина в выходные дни (Праздники не учитываются)

1. Подключаемся к нашей созданной ВМ: vagrant ssh
2. Переходим в root-пользователя: sudo -i
3. Создаём пользователя otusadm и otus: sudo useradd otusadm && sudo useradd otus
4. Создаём пользователям пароли: echo "Otus2022!" | sudo passwd --stdin otusadm && echo "Otus2022!" | sudo passwd --stdin otus

Для примера мы указываем одинаковые пароли для пользователя otus и otusadm

5. Создаём группу admin: sudo groupadd -f admin

6. Добавляем пользователей vagrant,root и otusadm в группу admin:
usermod otusadm -a -G admin && usermod root -a -G admin && usermod vagrant -a -G admin

Обратите внимание, что мы просто добавили пользователя otusadm в группу admin. Это не делает пользователя otusadm администратором.

После создания пользователей, нужно проверить, что они могут подключаться по SSH к нашей ВМ. Для этого пытаемся подключиться с хостовой машины:
ssh otus@192.168.57.10
Далее вводим наш созданный пароль.

```
➜  12_PAM ssh otus@192.168.57.10
The authenticity of host '192.168.57.10 (192.168.57.10)' can't be established.
ECDSA key fingerprint is SHA256:DF+i2KiB5/2eoWDEjljwbUyvPwVxbx+ZNSUjzMnqjV4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.57.10' (ECDSA) to the list of known hosts.
otus@192.168.57.10's password:
[otus@pam ~]$ whoami
otus
[otus@pam ~]$ exit
logout
Connection to 192.168.57.10 closed.
➜  12_PAM
➜  12_PAM ssh otusadm@192.168.57.10
otusadm@192.168.57.10's password:
[otusadm@pam ~]$ whoami
otusadm
[otusadm@pam ~]$ exit
logout
Connection to 192.168.57.10 closed.
➜  12_PAM
```

Если всё настроено правильно, на этом моменте мы сможем подключиться по SSH под пользователем otus и otusadm

Далее настроим правило, по которому все пользователи кроме тех, что указаны в группе admin не смогут подключаться в выходные дни:

7. Проверим, что пользователи root, vagrant и otusadm есть в группе admin:
[root@pam ~]# cat /etc/group | grep admin
printadmin:x:994:
admin:x:1003:otusadm,root,vagrant
[root@pam ~]#

Информация о группах и пользователях в них хранится в файле /etc/group, пользователи указываются через запятую.

Выберем метод PAM-аутентификации, так как у нас используется только ограничение по времени, то было бы логично использовать метод pam_time, однако, данный метод не работает с локальными группами пользователей, и, получается, что использование данного метода добавит нам большое количество однообразных строк с разными пользователями. В текущей ситуации лучше написать небольшой скрипт контроля и использовать модуль pam_exec

8. Создадим файл-скрипт /usr/local/bin/login.sh

vim /usr/local/bin/login.sh

```
#!/bin/bash
#Первое условие: если день недели суббота или воскресенье
if [ $(date +%a) = "Sat" ] || [ $(date +%a) = "Sun" ]; then
 #Второе условие: входит ли пользователь в группу admin
 if getent group admin | grep -qw "$PAM_USER"; then
       #Если пользователь входит в группу admin, то он может подключиться
       exit 0
     else
       #Иначе ошибка (не сможет подключиться)
       exit 1
   fi
 #Если день не выходной, то подключиться может любой пользователь
 else
   exit 0
fi
```

В скрипте подписаны все условия. Скрипт работает по принципу:
Если сегодня суббота или воскресенье, то нужно проверить, входит ли пользователь в группу admin, если не входит — то подключение запрещено. При любых других вариантах подключение разрешено.

9. Добавим права на исполнение файла: chmod +x /usr/local/bin/login.sh

10. Укажем в файле /etc/pam.d/sshd модуль pam_exec и наш скрипт:

vim /etc/pam.d/sshd

```
#%PAM-1.0
auth       substack     password-auth
auth       include      postlogin
auth required pam_exec.so debug /usr/local/bin/login.sh
account    required     dad
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open env_params
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    optional     pam_motd.so
session    include      password-auth
session    include      postlogin
```

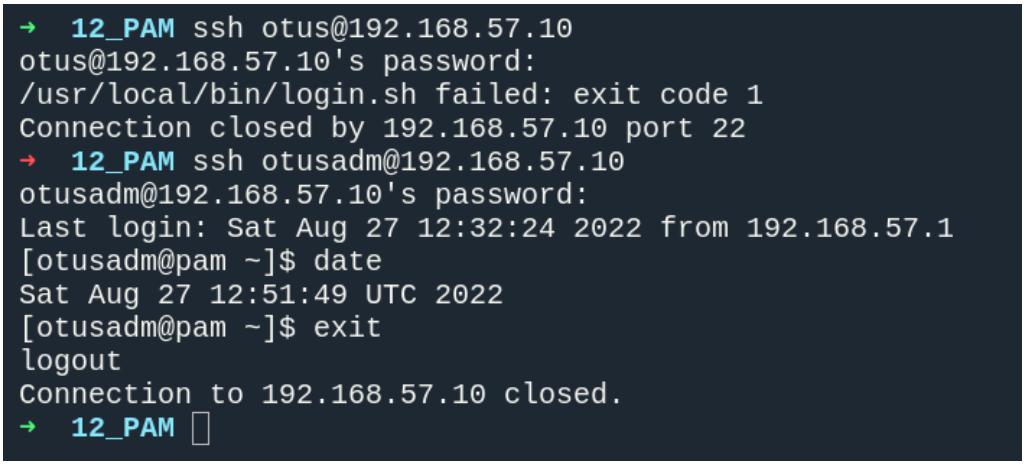
На этом настройка завершена, нужно только проверить, что скрипт отрабатывает корректно.

Если Вы выполняете данную работу в выходные, то можно сразу попробовать подключиться к нашей ВМ. Если нет, тогда можно руками поменять время в нашей ОС, например установить 27 августа 2022 года (Суббота):

sudo date 082712302022.00

и попробовать подключиться.

Если настройки выполнены правильно, то при логине пользователя otus у Вас должна появиться ошибка. Пользователь otusadm должен подключаться без проблем:



Рекомендуемые источники

- Статья об управлении пользователей