Sessão 5: Gestão de configuração

1) Instalação do Ansible

Clonar template para `ansible`, 10.0.42.5/24. Renomear e integrar no LDAP/SSH-ca como de costume. Criar entradas no DNS direto/reverso para `ansible.intnet`.

Como `root@ldap`, crie um usuário para o Ansible, membro dos grupos `setup` e `fwadm`:

```
# ldapadduser ansible setup
# ldapaddusertogroup ansible setup
# ldapaddusertogroup ansible fwadm
```

Como `root@nfs`, permita ao usuário `ansible` executar quaisquer comandos como `root` sem digitar senha:

```
# grep ansible /config/sudoers
ansible   ALL=(ALL:ALL)    NOPASSWD: ALL
```

Como `root@ansible`, instale o Ansible no servidor:

```
# echo "deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main" >
/etc/apt/sources.list.d/ansible.list
# apt-get install dirmngr
# apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367
# apt-get update
# apt-get install ansible
```

2) Execução de comandos simples

Como `ansible@ansible`, assine um par de chaves para logar nos servidores integrados no sistema LDAP/SSH-CA:

```
$ bash scripts/sshsign_user.sh
(sshca@10.0.42.2) Enter passphrase:
# 10.0.42.2:22 SSH-2.0-OpenSSH_7.4p1 Debian-10+deb9u4

(CA private key) Enter passphrase: seg10_user_ca
Signed user key id_rsa-cert.pub: id "ansible" serial 0 for ansible valid from 2018-11-
06T13:25:23 to 2021-11-05T12:30:23
```

Crie um diretório `~/ansible`, e um arquivo de hosts identificando as máquinas a gerenciar.

```
$ mkdir ~/ansible
$ cd ~/ansible
```

```
$ nano ~/ansible/hosts
(...)
```

```
$ cat ~/ansible/hosts
[srv]
fw
ldap
nfs
log
ansible
```

Execute um comando simples em todas as máquinas gerenciadas pelo Ansible.

```
$ ansible -i ~/ansible/hosts srv -b --become-user=root -m shell -a 'hostname ; whoami'
ansible | CHANGED | rc=0 >>
ansible
root

fw | CHANGED | rc=0 >>
fw
root

ldap | CHANGED | rc=0 >>
ldap
root

nfs | CHANGED | rc=0 >>
nfs
root

log | CHANGED | rc=0 >>
log
root
```

3) Uso de roles no Ansible

Vamos usar roles (papéis) no Ansible para configurar o sudo de forma local, mais segura. Crie o diretório `~/ansible/roles`, e inicialize o papel `sudoers`:

```
$ mkdir ~/ansible/roles
$ cd ~/ansible/roles/
```

```
$ ansible-galaxy init sudoers
- sudoers was created successfully
```

```
$ ls -R sudoers/
sudoers/:
defaults  files  handlers  meta  README.md  tasks  templates  tests  vars

sudoers/defaults:
main.yml

sudoers/files:

sudoers/handlers:
main.yml

sudoers/meta:
main.yml

sudoers/tasks:
main.yml

sudoers/templates:

sudoers/tests:
inventory  test.yml

sudoers/vars:
main.yml
```

Copie o arquivo `sudoers` do NFS para a pasta `files`:

```
$ cp /config/sudoers ~/ansible/roles/sudoers/files/
```

Observe as permissões do arquivo `sudoers` original. Com isso em mente, edite o arquivo `~/ansible/roles/sudoers/tasks/main.yml` como se segue:

```
$ ls -ld /etc/sudoers.old
-r--r----- 1 root root 669 jun  5  2017 /etc/sudoers.old
```

```
$ cat ~/ansible/roles/sudoers/tasks/main.yml
---
- name: Propagate sudoers configuration
  become: yes
  become_user: root
  copy:
    src: sudoers
    dest: /etc
    owner: root
    group: root
    mode: 0440
```

Crie o arquivo `~/ansible/srv.yml` para amarrar os hosts à nova role.

```
$ cat ~/ansible/srv.yml
---
- hosts: srv
  roles:
    - sudoers
```

Execute a role.

```
$ ansible-playbook -i ~/ansible/hosts ~/ansible/srv.yml

PLAY [srv] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [nfs]
ok: [ldap]
ok: [fw]
ok: [ansible]
ok: [log]

TASK [sudoers : Propagate sudoers configuration] *******************************
changed: [fw]
changed: [ldap]
changed: [nfs]
changed: [ansible]
changed: [log]

PLAY RECAP *********************************************************************
ansible                    : ok=2    changed=1    unreachable=0    failed=0
fw                         : ok=2    changed=1    unreachable=0    failed=0
ldap                       : ok=2    changed=1    unreachable=0    failed=0
log                        : ok=2    changed=1    unreachable=0    failed=0
nfs                        : ok=2    changed=1    unreachable=0    failed=0
```

Verifique que o arquivo `/etc/sudoers` é lido localmente, agora.

```
$ ls -ld /etc/sudoers
-r--r----- 1 root root 1392 nov  6 13:50 /etc/sudoers
```

4) Versionamento de configuração com git

5) sudo

1. Vamos configurar o arquivo `/config/sudoers` de acordo com a especificação da atividade. Usando o comando `visudo -f /config/sudoers`, edite o arquivo com o seguinte conteúdo:

```
 1 Defaults        env_reset
 2 Defaults        mail_badpass
 3 Defaults        secure_path
="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
 4
 5 User_Alias ADMINS    = aluno, \
 6                        luke,  \
 7                        han
 8
 9 User_Alias FWUSERS   = leia
10
11 User_Alias LDAPUSERS = %ldapadm
12
13 Host_Alias FWHOSTS   = fw
14
15 Host_Alias LDAPHOSTS = ldap
16
17 Cmnd_Alias FWCMDS    = /sbin/iptables
18
19 Cmnd_Alias LDAPCMDS  = /usr/sbin/ldapaddgroup,             \
20                        /usr/sbin/ldapadduser,              \
21                        /usr/sbin/ldapaddusertogroup,       \
22                        /usr/sbin/ldapdeletegroup,          \
23                        /usr/sbin/ldapdeleteuser,           \
24                        /usr/sbin/ldapdeleteuserfromgroup,  \
25                        /usr/sbin/ldapmodifygroup,          \
26                        /usr/sbin/ldapmodifymachine,        \
27                        /usr/sbin/ldapmodifyuser,           \
28                        /usr/sbin/ldaprenamegroup,          \
29                        /usr/sbin/ldaprenameuser,           \
30                        /usr/sbin/ldapsetpasswd,            \
31                        /usr/sbin/ldapsetprimarygroup
32
33 root      ALL=(ALL:ALL)    ALL
34
35 ADMINS    ALL=(ALL:ALL)    ALL
36
37 FWUSERS   FWHOSTS=(root)   FWCMDS
38
39 LDAPUSERS LDAPHOSTS=(root) LDAPCMDS
40
41 #includedir /etc/sudoers.d
```

O que estamos fazendo? Vamos ver:

- Nas linhas [5-10] definimos *aliases* (apelidos) de usuários para agrupar os elementos que serão configurados para usar o `sudo`. Criamos um *alias* `ADMINS` para agrupar os usuários `aluno`, `luke` e `han`, `FWUSERS` para `leia` e `LDAPUSERS` para o **grupo** `ldapadm`. É especialmente importante manter um *alias* apontando para um usuário local, como o usuário `aluno`, caso

haja problemas com o LDAP.

- Nas linhas [12-14] definimos *aliases* para máquinas, `ns1` e `ns2`. Também poderíamos usar endereços IP, se desejado.
- Nas linhas [16-30] definimos *aliases* de comandos: para a máquina `ns1`, apenas o comando `/sbin/iptables` é suficiente; já para a máquina `ns2` configuramos uma lista detalhada dos comandos que o *alias* `LDAPUSERS` poderá usar.
- Nas linhas [32-38] fazemos a "amarração" dos *aliases* previamente definidos, atribuindo aos usuários/grupos em quais máquinas eles podem executar os comandos, como quais usuários, e quais são esses comandos.

2. Vamos testar o acesso de `leia` na máquina `ns1`. Antes disso o primeiro passo, é claro, é criar o diretório `/config` e configurar sua montagem automática durante o *boot* via `/etc/fstab`. Acesse `ns1` como `root`, crie o diretório `/config` e insira a linha a seguir no final do arquivo:

```
# hostname ; whoami
fw
root
```

```
# mkdir /config
```

```
# nano /etc/fstab
(...)
```

```
# tail -n1 /etc/fstab
10.0.42.3:/config /config nfs defaults 0 0
```

Monte o diretório e verifique seu conteúdo:

```
# mount -a
```

```
# mount | grep config
10.0.42.3:/config on /config type nfs4
(rw,relatime,vers=4.2,rsize=131072,wsize=131072,namlen=255,hard,proto=tcp,port=0,ti
meo=600,retrans=2,sec=sys,clientaddr=10.0.42.1,local_lock=none,addr=10.0.42.3)
```

```
# ls /config/
sudoers
```

Agora, renomeie o arquivo `/etc/sudoers` e crie o link simbólico:

```
# mv /etc/sudoers /etc/sudoers.old ; ln -s /config/sudoers /etc/
```

Perfeito, agora vamos testar o funcionamento da configuração. Como `leia`, tente executar o comando `iptables` usando o `sudo`:

```
$ hostname ; whoami
fw
leia
```

```
$ sudo iptables -L
[sudo] senha para leia:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Excelente! E se tentarmos executar um comando não autorizado?

```
$ sudo rm /etc/shadow
Sinto muito, usuário leia não tem permissão para executar "/bin/rm /etc/shadow"
como root em fw.intnet.
```

De fato, é possível listar exatamente quais comandos um usuário está apto a executar com o comando `sudo -l`:

```
$ sudo -l
Entradas de Defaults correspondentes a leia em fw:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin


Usuário leia pode executar os seguintes comandos em fw:
    (root) /sbin/iptables
```

E quanto a `han`? Ele consegue executar qualquer comando como `root`?

```
$ hostname ; whoami
fw
han
```

```
$ sudo -l
[sudo] senha para han:
Entradas de Defaults correspondentes a han em fw:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin


Usuário han pode executar os seguintes comandos em fw:
    (ALL : ALL) ALL
```

Perfeito! A última questão é a seguinte: e se `leia`, por qualquer motivo, conseguir obter a senha do usuário `root`? O que não é exatamente difícil, já que estamos usando `rnpesr` como senha. Nesse caso, ela terá acesso irrestrito:

```
$ hostname ; whoami
fw
leia
```

```
$ su -
Senha:
```

```
# whoami
root
```

A solução ideal, nesse caso, é desabilitar a senha do `root`. Com isso, mesmo que os usuários saibam a senha, ela não poderá ser usada para efetuar escalada de privilégios usando o `sudo`. Podemos usar o comando `passwd -l` para fazer isso:

```
# passwd -l root
passwd: informação de expiração de senha alterada.
```

```
# exit
```

```
$ whoami
leia
```

```
$ su -
Senha:
su: Falha de autenticação
```

Com a senha desabilitada, apenas aqueles usuários que tenham permissão de `sudo` para executar comandos de escalada de privilégio poderão tornar-se o usuário `root` — todos os demais, restritos a um subconjunto de comandos controlados pelo arquivo `/config/sudoers`, não conseguirão fazê-lo.

Note que mesmo o usuário `han`, que possui acesso irrestrito, não consegue executar `su` diretamente:

```
$ whoami
han
```

```
$ su -
Senha:
su: Falha de autenticação
```

```
$ sudo --login
```

```
# whoami
root
```

Apenas via `sudo su` ou `sudo --login` (que equivale a invocar um *shell* de login, como executar `sudo bash`) é possível escalar privilégio, como demonstrado.

A leitura do arquivo `/config/sudoers` a partir de um compartilhamento de rede, via NFS, traz consigo uma preocupação de segurança bastante relevante — e se a máquina `nfs` estiver indisponível? Com efeito, se isso acontecer teremos grandes problemas, já que toda a configuração de autorização do sistema local estará indisponível. Por esse motivo, é fundamental que o `sudoers` esteja acessível localmente, o que faremos na sessão 6 deste curso.

Por ora, vamos torcer para que nada catastrófico aconteça com a máquina `nfs`. Dedos cruzados.

3. Vamos para o caso do usuário `chewie`. Acesse a máquina `ns2` como o usuário `root` e:
   - Crie o diretório `/config`.
   - Configure sua montagem automática durante o *boot* via `/etc/fstab`.
   - Configure o `sudo` para ler a configuração do `/config/sudoers`.
   - Desabilite a senha do usuário `root`.
   - Teste o funcionamento da configuração com os usuários `chewie` e `luke`.

Dada a semelhança dos primeiros quatro itens com o passo anterior, iremos passar diretamente para o passo final, assumindo que o aluno completou a configuração com sucesso.

Como o usuário `chewie` na máquina `ns2`, verifique quais comandos você está autorizado a executar usando o `sudo`:

```
$ hostname ; whoami
ns2
chewie
```

```
$ sudo -l
Entradas de Defaults correspondentes a chewie em ldap:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin


Usuário chewie pode executar os seguintes comandos em ldap:
    (root) /usr/sbin/ldapaddgroup, /usr/sbin/ldapadduser,
/usr/sbin/ldapadduserto group,
        /usr/sbin/ldapdeletegroup, /usr/sbin/ldapdeleteuser,
/usr/sbin/ldapdeleteuserfromgroup,
        /usr/sbin/ldapmodifygroup, /usr/sbin/ldapmodifymachine,
/usr/sbin/ldapmodifyuser,
        /usr/sbin/ldaprenamegroup, /usr/sbin/ldaprenameuser,
/usr/sbin/ldapsetpasswd,
        /usr/sbin/ldapsetprimarygroup
```

Tente criar um novo grupo no LDAP, `sudotest`, e em seguida delete-o.

```
$ sudo ldapaddgroup sudotest
Successfully added group sudotest to LDAP
```

```
$ sudo ldapdeletegroup sudotest
Successfully deleted group cn=sudotest,ou=Groups,dc=intnet from LDAP
```

Tente executar um comando não-autorizado:

```
$ sudo reboot
Sinto muito, usuário chewie não tem permissão para executar "/sbin/reboot" como
root em ldap.intnet.
```

Como `luke`, tente logar diretamente como o `root` usando o `su`.

```
$ hostname ; whoami
ns2
luke
```

```
$ sudo su -
```

```
# whoami
root
```

4. A máquina `nfs` já está praticamente configurada — a pasta `/config` é local, o que dispensa a montagem automática durante o *boot*, e o `/config/sudoers` já foi configurado e testado nos passos (2) e (3). Resta apenas desabilitar a senha do `root` — faça isso:

```
# hostname ; whoami
nfs
root
```

```
# passwd -l root
passwd: informação de expiração de senha alterada.
```

5. Idealmente, seria interessante que novas máquinas derivadas da VM `debian-template` estivessem automaticamente integradas com o sistema de `sudo` centralizado que acabamos de configurar nesta atividade. Para isso, vamos fazer algumas alterações rápidas na máquina.

   No Virtualbox, com a máquina desligada, em *Settings* > *Network* > *Adapter 1* > *Attached to*, escolha *Host-only Adapter*. O nome da rede *host-only* deve ser o mesmo alocado para a interface de rede da máquina virtual `ns1`, configurada durante a sessão 2, que está conectada à DMZ.

   Ligue a máquina `debian-template`, e acesse como o usuário `root`.

   Reconfigure a rede em `/etc/network/interfaces` para a DMZ, com o endereço IP 10.0.42.250/24:

```
# hostname ; whoami
debian-template
root
```

```
# nano /etc/network/interfaces
(...)
```

```
source /etc/network/interfaces.d/*

auto lo enp0s3

iface lo inet loopback

iface enp0s3 inet static
address 10.0.42.250/24
gateway 10.0.42.1
```

Crie a pasta `/config` e configure sua montagem automática no arquivo `/etc/fstab`:

```
# mkdir /config
```

```
# echo "10.0.42.3:/config /config nfs defaults 0 0" >> /etc/fstab
```

Configure o *symlink* do arquivo `/etc/sudoers`:

```
# mv /etc/sudoers /etc/sudoers.old ; ln -s /config/sudoers /etc/
```

Finalmente, desabilite a senha do usuário `root` — usaremos o `sudo` com o usuário `aluno` para efetuar a configuração inicial das novas máquinas derivadas da VM `debian-template`:

```
# passwd -l root
passwd: informação de expiração de senha alterada.
```

Desligue a VM `debian-template`.