laSalle ENG
Universitat Ramon Llull

Àrea d'Informàtica

Administració i Disseny de Sistemes

Data lliurament: 18 de Novembre de 2018 **Curs:** 2018/2019

Objectius de la pràctica

Aquesta pràctica d'Administració i Disseny de Sistemes té com a objectius:

- Aplicar en un escenari real els coneixements adquirits en l'assignatura.
- Implementar polítiques de seguretat.
- Aprendre a realitzar i implementar mecanismes de comunicacions entre processos.
- Modificar el codi font i recompilar paquets de *software*.
- Contenidorització d'usuaris mitjançant la creació d'entorns.
- Entendre i aplicar l'estructura de l'estàndard *FHS* per a distribucions Linux.
- Gestionar els processos i serveis que s'executen dins del sistema.

RBAC: *Rol-based access control*

Des del departament de Sistemes de la Salle, es vol actualitzar el sistema de gestió d'usuaris dels servidors interns i han dissenyat una nova estructura basada en rols, és a dir, cada usuari accedirà remotament en un entorn individualitzat i restringit, caracteritzat pel rol assignat.

Després del disseny, es varen adonar compte que no el podien desenvolupar, fet donat pel desbordament de noves peticions d'alta de nous usuaris aquest any. És per això, que han delegat el desenvolupament als alumnes cursant *Administració i Disseny de Sistemes*.

Tenint en compte que es tracti d'un sistema complex, s'ha pensat el desenvolupament en **dues fases separades**, però que el bon funcionament de la primera és elemental per la segona.

Fase 1 - *The proof of concept*

Aquesta primera esdevindrà la prova i base pel RBAC, és a dir, s'utilitzarà una imatge d'una distribució de Linux en concret i es modificarà per satisfer tots els requeriments. Un cop validat el seu correcte funcionament, la segona part tractarà sobre la generalització i estructuració d'aquest sistema, per a la seva distribució com a aplicació, que sigui capaç de desplegar-se en imatges totalment noves.

A continuació es detallarà punt a punt, els requeriments que cal satisfer estrictament en aquesta fase:

- Ha d'utilitzar la distribució de Linux **Ubuntu Server 18.04.1 LTS**, sense afegir cap interfície gràfica.
- Per a facilitar el traspàs del entorn de desenvolupament, és necessari executar-ho en una màquina virtual utilitzant un *hypervisor* com *VirtualBox* o *VMware Workstation*.
- **Estructuració dels scripts i fitxers**. S'ha de desar tant els fitxers de configuració com els scripts, seguint l'estàndard `FHS` (*Filesystem Hierarchy Standard*) [1].
- **SSH**. La connexió remota es farà mitjançant SSH, amb les següents funcionalitats:
  - L'autentificació dels usuaris es realitzarà mitjançant una *Key*[2] i amb el sistema 2FA (utilitzant el mòdul de PAM *libpam-google-authenticator*). Per aquest motiu, cap usuari ha de tenir contrasenya.
  - **[Opcional]** El rol DataStore només tindrà *2FA* i el Visitor *Key*, els dos mètodes pels altres.
  - A més a més, no s'ha de poder accedir mitjançant l'usuari `root` de manera remota.
- **Dades**. Les dades dels usuaris com la seva gestió ha de seguir:
  - Tant els desplegaments dels entorns (no els scripts que els generen) com les dades dels usuaris, han d'estar en un **disc a part**, no una partició del mateix disc, muntat en el directori principal del sistema. Seguint la ruta per a desar les dades: `/users/<rol>/<user_name>`.
  - El muntatge ha de ser persistent, no ha de ser muntat amb un script en iniciar el sistema.
- **Creació d'usuaris**. S'haurà de crear un *script* per tal d'afegir i treure usuaris, que permeti **només** a l'administrador del sistema, executar en qualsevol moment la comanda `$ rbac [commanda]` per dur a terme les gestions necessàries. A més a més, ha de contenir una opció de `help`, que mostri com s'utilitza.

  Cal remarcar que el rol que es passi en el moment de la creació de l'usuari, en cas de no existir, es crearà sempre que existeixi un fitxer de configuració per aquest. Cal tenir en compte, que aquest procés ha de ser confirmat abans de procedir i en cas de no trobar-se el fitxer, s'aturarà la creació.
- **Fitxer base de configuració**. En aquest fitxer s'haurà d'indicar:
  - Directori que conté els fitxers de configuració dels rols.
  - Correu de l'administrador del sistema, per qualsevol incidència o petició.
  - Configuracions per a IPC (*interprocess communication*), com adreça, ports o similars.
  - Finalment, qualsevol configuració que es vegi necessària pel correcte funcionament del RBAC.

[1] https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html
[2] https://help.ubuntu.com/community/SSH/OpenSSH/Keys

- **Entorns d'usuari**. El sistema ha de complir els següents punts:
  - Han de ser creats de manera automàtica a partir d'un script, no es pot crear una base i fer el `cp` per a les noves connexions. A més a més, s'ha d'utilitzar PAM com a mètode d'execució del script.
  - Per aquesta primera implementació, les versions dels *binaries* seran la mateixa que la del S.O. pare.
  - En el moment de la creació, es a dir, quan es connecti l'usuari al sistema i no estigui ja creat, es llegirà un fitxer de configuració depenent del rol que tingui, que definirà:
    * Programes disponibles en l'entorn.
    * Localització del fitxer `.bashrc` a utilitzar i els que cregueu necessaris pel bon funcionament.
    * Dues configuracions numèriques, primer indicant els dies que l'entorn estarà disponible (per defecte només durant connexió) i si la `home` és persistent, o bé té un temps en dies definit (per defecte indefinit).
    * Des de la primera connexió es defineix la data límit. Un cop superat i efectivament executat el programa de neteja, es torna a definir en la següent connexió.
    * L'eliminació dels entorns o directoris `home`, es realitzarà sempre a les 00.00 del dia límit.
    * En cas que l'usuari estigui connectat s'aplacarà pel dia següent. [**Opcional**] Tan bon punt es desconnecti.
  - L'usuari disposarà del *path* `<path_al_entorn>/home/<user_name>` per als seus fitxers.
  - S'ha d'utilitzar `chroot` per mantenir els usuaris, recomanable usar el servei de *SSH* per realitzar-ho.
  - En cas de trobar un error durant la creació, ha d'aturar-se i informar correctament sobre l'error a qui anava dirigit l'entorn.
- **Rols d'usuaris**. Els rols d'usuaris han de seguir estrictament l'explicació en l'apartat a continuació.

  1. **DataStore**. Aquest usuari serà pensant únicament per desar dades. No està pensat per ser capaç d'executar cap comanda, ja que només utilitzarà la connexió per pujar fitxers mitjançant el protocol *sftp*.

     Donat la raó de la seva existència, l'entorn haurà de contenir únicament el *path* a la seva *home*. Hom pot pensar que no té cap sentit mantenir l'estructura de `<path_al_entorn>/home/<user_name>`, però ens han fet entendre que a vegades aquest pot pujar de categoria i d'aquesta manera, serà fàcilment adaptable a una superior.

  2. **Visitor**. El següent nivell jeràrquic es tracta del *Visitor*. Com bé el seu nom indica, l'objectiu d'aquest és per crear entorns temporals, tant per proves, com per permetre l'ús del servidor en cursos de curta durada. Per aquest motiu, aquests tindran una configuració on l'entorn i la `home` s'eliminaran passat el dia (per defecte, ja que es podrà modificar segons el contingut del fitxer de configuració).

     Com que aquest grup pot tenir diferents entorns caracteritzats per un tipus de comanda dependent del seu ús, tots els usuaris que utilitzin aquest grup hauran de seguir la següent nomenclatura:

     $$< objectiu > \_ < nombre >$$

     Així per exemple, si es volgués crear un grup d'usuaris per un curs determinat del màster, seguirien l'estructura: `master-curs057_001`, `master-curs057_002`, etc.

     Tal com s'ha comentat, aquest no només tindrà un únic entorn, sinó que dependrà de l'objectiu de l'usuari, que contingui un o altre. És per això, que tot i que aquest tindrà una estructura bàsica, a la vegada si es troba un fitxer de configuració per l'objectiu, és a dir, que tingui com a nom la primera part del nom de l'usuari separat amb '_', carregarà l'entorn basat en aquest. Només en casos que no s'especifiqui cap configuració per objectiu, aquest contindrà per defecte:

     (a) bash
     (b) touch & mkdir & rm
     (c) cd
     (d) ls & ll
     (e) vim & nano

És important, que en el cas de `bash`, s'haurà de modificar el missatge "*: command not found*"pel "*El teu rol no disposa de la comanda: *", tant per aquest rol, com per als següents. Aquesta modificació però, no haurà de dependre de cap fitxer de configuració extra que modifiqui el missatge com el *.bashrc*.

3. **Basic**. El primer usuari no temporal que contindrà comandes es tracti del *Basic*. Per aquests usuaris, l'entorn haurà de durar el dia de connexió, però la `home` serà persistent. Aquest mantindrà les mateixes comandes que el visitant afegint:
   (a) gcc
   (b) make
   (c) kill

   Està pensat principalment per usuaris que pertanyen al primer curs, que només necessiten compilar i desenvolupar programes en `C`.

   Cal remarcar, que aquest usuari serà la base pels següents, per tant, **totes les noves comandes que els rols superiors tinguin, seran juntament amb la llista dels rols anteriors** (e.g. Medium té Medium + Basic + Visitor).
4. **Medium** Seguidament, tenim l'usuari *Medium*. Són usuaris que ja comencen a tenir un coneixement més elevat d'un entorn Linux i per tant, se'ls hi proporciona més eines de desenvolupament.
   (a) java
   (b) ln
   (c) ps
   (d) python & pip/pip3
   (e) valgrind
   (f) grep
   (g) awk
   (h) sed

   Com en el rol anterior, l'entorn pels usuaris d'aquest rol s'eliminarà cada dia, però les dades personals hauran de romandre persistents indefinidament.
5. **Advanced** Aquests es tracten d'usuaris que tot i no tenir permisos d'administrador en el sistema pare, tindran permisos per afegir i modificar el seu entorn. D'aquesta manera, podran compilar i afegir els paquets que vulguin.

   Per facilitar-los la gestió i anàlisi, seran capaços d'executar:
   (a) chmod, chown
   (b) strace
   (c) 3 més, que cregueu que siguin necessaris, per poder administrar correctament l'entorn.

   Finalment, els usuaris que pertanyen a aquest rol, se'ls hi proporcionarà un entorn complet (incloent-hi la `home`) indefinit.

- **Programa de gestió de l'entorn**. S'haurà de crear un programa mitjançant l'interpret `bash` que permetrà gestionar l'entorn amb les següents comandes:
  - `clean-all`: Eliminar l'entorn i la `home`. Es demanarà dues confirmacions, primer confirmant amb S/N i després es demanarà escriure el nom d'usuari. Un cop eliminats, es tornarà a crear un nou seguint el rol de l'usuari.
  - `reset`: Eliminar i tornar a carregar l'entorn, amb una confirmació de S/N, sense eliminar les dades del usuari, es a dir, la *home*.
  - `help`: Mostrarà les comandes disponibles del programa amb una explicació del seu ús.
  - `list-commands`: Mostrarà les comandes disponibles a executar (e.g. ln, pyhton, ps, make, etc.).
  - `request-command "[missatge]"`: S'enviarà el missatge al correu configurat en el fitxer base.

  El programa haurà d'executar-se mitjançant `$ environment [commanda]`

Consideracions

1. La realització i entrega d'aquesta pràctica és en grups de 2 persones.

2. Es realitzarà una entrevista presencial per tots els membres del grup, en la qual es realitzaran proves sobre la implementació. No passar l'entrevista, significa que s'haurà d'entregar de nou la practica amb una màxima nota inferior.

3. El mètode d'entrega d'aquesta pràctica es farà mitjançant la pujada amb el format `login1_login2.zip` de tots els fitxers necessaris per poder executar aquesta pràctica en una imatge neta de *Ubuntu Server 18.04.1* (sense incloure executables/fitxers del sistema que no hagin estat modificats). Per aquest motiu, haurà de seguir una estructuració en format `FHS`.

   En aquest *zip* haurà d'estar també un `README`, explicant breument la funcionalitat que ofereix cada script/fitxer de creació vostre en el sistema **RBAC**.

4. A més a més, **s'haurà de fer un control de versions en una plataforma Git** durant tot el desenvolupament (Github o Gitlab) per revisar de cara a l'entrevista.