C02-script-user_advenced-variable-vim - TP LINUX - MTN

Guillaume ASTIER

26/02/16

Table des matières

ISEN - C02_UNIX - TP n°2

- Gestion des utilisateurs, edition et scripting.
- Objectifs : Prise en main de l'édition / scripting / Gestion des users sous PAM
- Tous ces exercices doivent être faits en mode commande dans un shell

Gestion des utilisateus sous PAM

- Editer le fichier " '/etc/shadow' ".
- Quel est la fonction de hashage utilisée pour chiffrer les mots de passe ? (chercher dans les pages de man : crypt)
- Quel fichier de configuration permet de changer la méthode de chiffrement des mots de passe ?
- Modifier ce fichier pour que les nouveaux mots de passe soient désormais chiffrés en 'sha256'.
- Changer le mot de passe de "eleve1" afin de vérifier la prise en compte de la modification.

Sécurité et brute force

- A l'aide de la commande "john" (" 'man john' "), vérifiez la robustesse des mots de passe.
- Regarder les pages " 'man' " des appels systèmes " 'getuid' ", " 'geteuid' ", " 'setuid' ".

Compilation de source

- Écrire un programme qui affiche à l'écran les UID réel et effectif, et le résultat de la commande " 'id' ".

Pour ceux qui n'ont pas de notion en langage C vous pouvez récupérer le code source sur : http://isen.gastier.net/RESSOURCE/source.c

Pour compiler un programme en language C sous linux on utilise gcc

exec / stickybit

- Mettre "root" comme utilisateur propriétaire du fichier binaire, et positionner la permission setuid.
- Exécuter votre programme en "eleve1". La commande " 'id' " est-elle exécutée avec des droits "root" ? Si vous répondez non, améliorer le programme pour que id affiche " 'uid=0(root), gid=0(root)' ".
- Modifier le programme pour qu'il permette à un utilisateur de lancer un shell root, sans qu'aucune authentification ne soit nécessaire. l'aide se trouve dans le man cf 2.2

Création de fichier

- Trouvez 3 moyens de populer/editer un fichier (peut imorte l'outils)
- Créez le fichier 'stat_local.sh' vide dans /home/isen/
- Editez le avec vim et ajouté le shebang associé à un script bash

Arguement

- Ajoutez le code nécessaire pour que 3 arguments soient passés au script.Ce dernier doit nous retourner après execution le nom des 3 arguments et leur contenue
- A l'aide de la commande ''wc' ' et ses options, indiquez le nombre de carcatères contenuent dans chaque variable
- Ajoutez avant l'affichage une ou plusieurs conditions que le script à bien reçu 3 arguments. Si ce n'est pas le cas, le script doit s'arrêter avec un message

Sous commande

- A chaque lancement vous devez récupérer la date et l'heure de lancement au format YYYYMMDD HHMMDD dans une variable
- Grâce à la varible qui contient la date et en utilisant la commande logger vous devez populer le fichier /var/log/'stat_local'

ajout fonction

- Crééz 3 fonctions qui doivent :
  - fonction 1 : récupère l'arguement 1 et crée le fichier du même nom dans /tmp/
  - fonction 2 : récupère l'arguement 2 si ce dernier est égale à 'y' modifie l'accès à tous du fichier de la fonction 1
  - fonction 3 : récupère l'arguement 3 qui doit être le path d'un intérpréteur (bash / bin / perl / python)

Inception

- Dérnière partie du script. Le fichier de l'argument 1 a été créé. les droit sont en éxécution.
  - ajouter en header le shebang associé à l'intérpréteur passé dans 'ajout fonction [3]'. Aidez vous de which (man which)
  - Le script devra boucler en vous demandant "quel commande voulez vous ajouter à votre $script". $script sera l'arguement 1.
    - *Tant que le script ne récupère pas la chaine ENDEDIT il doit continuer à vous poser la question et intégrer en fin de fichier la chaine passé dans l'arguement.*
    - *Les commande à passer en arguement de la boucle sont : whoami ; pwd ; id ; df ; du -hs .*
  - Et enfin une fois sortie de la boucle le script devra vous demander "voulez vous execture votre script $script" et le lancer.

Figure 1: FIN DU TP 1