Cours de Sécurité Informatique
-
LNX 1

Sylvain Leroy - sylvain@eternilab.com
Jérôme Maurin - jerome.maurin@eternilab.com
Grégory Kirijean - gkirijean@gmail.com

December 1, 2020

Contents

Chapter 1

LNX1

1.1 License

1.2 Introduction

Tous les jours, vous utilisez des systèmes d'exploitation sans en prendre conscience. Vous leur confiez des données personnelles comme vos photos, vos numéros de carte bancaire, numéros de sécurité sociale, vos mots de passe, ... Pour que ces données garde leur confidentialité, leur intégrité et leur disponibilité, le système qui les manipule a un besoin de protéger ces données contre de nombreux types d'attaques.

Dans ce cours, nous allons voir comment sécuriser un système GNU/Linux pour pouvoir protéger au mieux les données qu'il manipule.

L'ANSSI (Agence nationale de sécurité des systèmes d'information) propose un guide de "Recommandations de sécurité relatives à un système GNU/Linux" (BP-028) que vous pourrez trouver ici : https://www.ssi.gouv.fr/uploads/2016/01/linux_configuration-fr-v1.2.pdf

Ce guide sera à utiliser pendant ce TP.

1.3 Instructions pour le rendu

Voici ce que vous devez faire pour réaliser votre rendu :

1. L'ensemble du rendu devrait être validé sur un système Debian dans sa version stable.
2. Si une liste de fonctions ou d'outils vous est donnée au début d'un exercice, vous ne pouvez utiliser QUE ces fonctions/outils durant cet exercice.
3. Si aucune fonction ou outil ne vous est donné pour un exercice, vous êtes libre d'utiliser ce que vous voulez dans le respect des langages du TP.
4. Le signe $ dans les exemples représente la ligne de commande dans laquelle vous pouvez entrer des commandes.
5. Vous DEVEZ rendre une tarball propre, en utilisant le modèle suivant pour son nommage : *nomdefamille*.tar.bz2
6. Vous pouvez réaliser ce projet par groupe de 2.
7. Si vous réalisez ce travail à plusieurs, vous devez écrire le nom de chaque personne dans le fichier AUTHORS.
8. En cas de doute, vous DEVEZ envoyer un courriel pour poser une question. Si un élément n'est pas clair et qu'aucune question n'a été posée, la décision finale appartient à l'enseignant. Aucune discussion ne sera possible.
9. Tous les courriels que vous envoyez devront respecter les règles suivantes :
   - L'objet du courriel doit être exactement : `[X][LNX1][Y]nomdefamille`
     - X correspondant au code de votre promotion
     - Y correspondant à HANDIN ou QUESTION
   - le corps du courriel doit être écrit dans un français sans faute

   Tout courriel ne respectant pas ces critères ne sera pas traité.

La tarball que vous devez rendre doit contenir EXACTEMENT la même sortie que la commande *tree* suivante (exception du *nomdefamille* que vous devez adapter) :

```
nomdefamille/
|-- ANSWERS
|-- AUTHORS
|-- ex1
|   |-- libpwquality
|   |   |-- common-password
|   |-- passwdqc
|       |-- common-password
|-- ex2
|   |-- usr.bin.ls
|-- ex3
|   |-- aide
|   |-- aide.conf
|   |-- aide.db
|-- ex4
|   |-- esiee.prf
|   |-- plugin_esiee
|-- README

6 directories, 11 files
```

- Le fichier ANSWER contient vos réponses aux questions posées dans le sujets, s'il y en a. S'il n'y a pas de question, vous devez laisser un fichier vide.
- Le fichier AUTHORS contient le nom des auteurs du rendu, chaque ligne contenant : 'prénom nom" suivi d'un retour à la ligne d'un des auteurs.
- Le fichier README ne contient rien, vous devez laisser un fichier vide.

1.4 Comment sécuriser un système ? Grands principes

Pour protéger ces données, il faut commencer par identifier le besoin en sécurité du système. Ainsi on va se poser plusieurs questions :

- Quelles sont les données à protéger ?
- Certaines données manipulées sont-elles plus critiques que d'autres ?
- Y a-t-il du matériel physique dont l'accès est critique autre que du stockage de données ? (https://fr.wikipedia.org/wiki/Stuxnet, super-calculateur, ...)
- Quel est le "besoin(/droit) d'en connaître" des utilisateurs ?

Parcourir la section 3 "Principes généraux de sécurité et de durcissement" du guide ANSSI BP-028.

1.5 Peut-on trop sécuriser un système ?

Trop de sécurité tue la sécurité :

- Les utilisateurs sont enclins au contournement lorsque la sécurité alourdit trop leur charge de travail

Il faut :

- Travailler avec l'utilisateur et l'accompagner dans le changement.
- Éviter une sur-sécurisation qui entraîne une gène utilisateur.

1.6 Méthodologie d'authentification : Approche multi-facteurs

Dans le domaine de l'authentification, pour augmenter le niveau de sécurité on utilisera une combinaison de techniques.
Cette approche est appelée "multi-facteurs".

On peut les séparer en trois catégories :

- Savoir :
  - Mots de passe
  - Réponse à des questions, ...
- Posséder :
  - Application mobile (attention à l'auto-usage)
  - Token physique : Yubikey, Nitrokey, Ledger, ...
- Être
  - Biométrie : Digital, optique, ...

La stratégie sera à adapter en fonction du besoin de sécurité, basée sur un ou plusieurs facteurs.
Parcourir la section 6.4 "PAM et NSS" du guide ANSSI BP-028.

1.7 Firefox dans la vie de tous les jours

Dans votre vie de tous les jours, il faut faire attention lorsque vous allez sur internet. Pour cela, il faut correctement configurer votre navigateur internet.

Voici quelques extensions qui pourraient vous être utiles :

- HTTPS Everywhere : force le https
- uBlock Origin (ou un autre adBlock)
- Privacy Badger : bloque les traceurs
- NoScript : bloque l'exécution du javascript
- Decentraleyes : bloque les traceurs et mouchards en complément des autres extensions et protège la confidentialité.
- Facebook container : isole Facebook dans une sous-partie de Firefox dédiée pour limiter le suivi via Facebook

Ensuite dans la configuration de votre navigateur, veillez à bloquer les cookies, traceurs, les fenêtres pop-up ... (c'est dans l'onglet sécurité des paramètres de firefox).

1.8 Exercice 1 : PAM

PAM est un mécanisme qui permet de fournir et spécifier de l'authentification sur un système *nix.

Dans cet exercice il vous faudra vous familiariser avec cette technologie et la prendre en main à travers quelques exercices.

Vous devez mettre en place un mécanisme qui impose que les mots de passe du système soient au moins de 9 caractères et qui contiennent majuscules, minuscules, chiffres et caractères spéciaux (donc un de chaque au moins).

Vous devez réaliser cet exercice en utilisant :

1. la bibliothèque PAM : *libpwquality*
2. la bibliothèque PAM : *passwdqc*

Pré-requis Installation de ces 2 bibliothèques PAM : Pour ce faire, il faut installer les paquets `libpam-pwquality` et `libpam-passwdqc`.

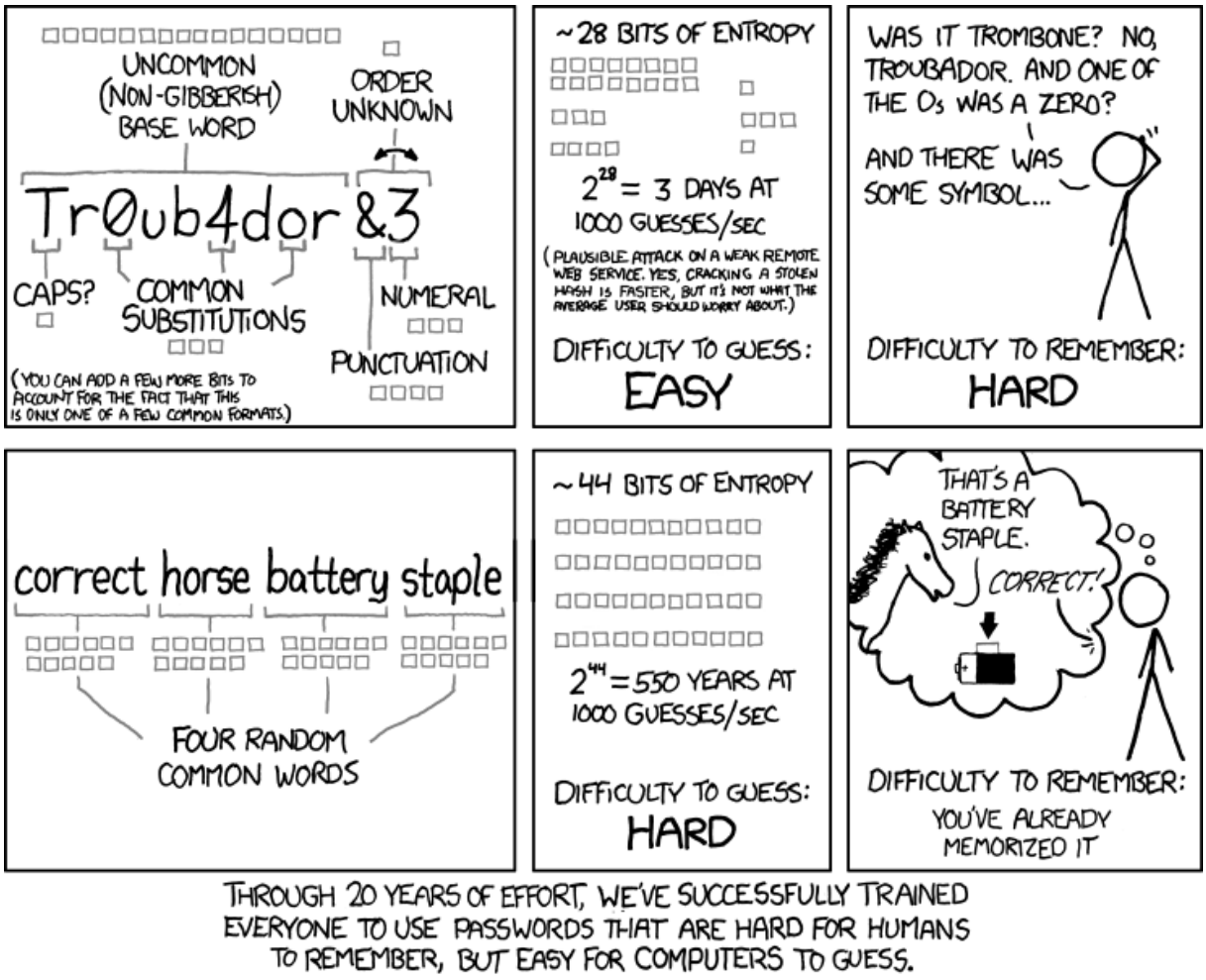


Figure 1.1: Source : https://xkcd.com/936/

Attention Gardez toujours une session ouverte lorsque que vous configurez PAM.

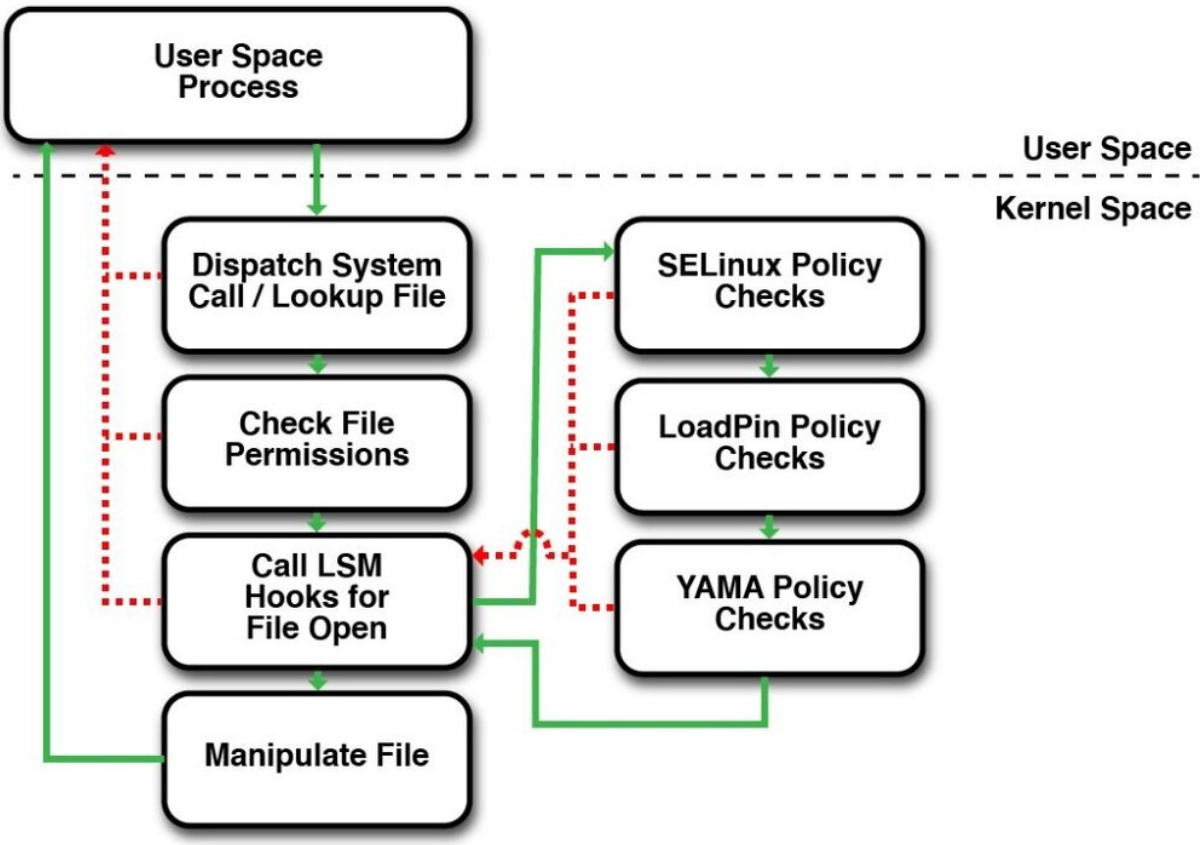
1.9 Contrôle d'accès

1. Théorie
    - DAC (Discretionary Access Control) : Le propriétaire décide
    - MAC (Mandatory Access Control) : Une autorité décide
    - RBAC (Role-Based Access Control) : Une liste de rôle décide
2. Pratique
    - Droits d'accès *nix, utilisateur, groupe, autres, ...
        - Commandes : adduser, moduser, chmod, chown, passwd ...
        - Fichiers : /etc/passwd, /etc/shadow, /etc/group, /etc/security/*, ...
    - *Capabilities* Linux : Droits détaillées d'actions possibles pour les processus (cf : man 7 capabilities)
        - Commande : setcap
    - ACLs (Access Control List) : Droits étendus par rapport aux classiques *nix
        - Commandes : setfattr, getfattr, setfacl, getfacl
        - Contraintes : le FS doit supporter les ACLs ou les attributs étendus. De plus, l'option doit être spécifier au montage du FS.
    - LSM (Linux security modules) permet un contrôle d'accès avancé :
        - SElinux (Sponsorisé par la NSA et RedHat. Très/trop compliqué.)
        - Apparmor (standard Debian 10)
        - ...
    - ...

Parcourir la section 6.5 "Vérification systèmes de fichiers et droits" et 7.2 "contrôle d'accès et mécanismes de sécurité avancés" du guide ANSSI BP-028.

1.9.1 SELinux LSM path

Voici un exemple de chemin de donné utilisé dans la gestion des droits fait par SELinux :



1.10 Exercice 2 : Apparmor

Vous devez mettre en place une configuration apparmor qui vous empêche de lister le contenu du répertoire /boot (et uniquement lui, pas ses sous-répertoires) avec la commande ls.

Pré-requis

1. Copier /bin/ls dans /usr/local/bin/ls pour éviter de se bloquer et pour gérer le cas où apparmor protège implicitement le binaire ls.
2. Vérifier que /usr/local/bin est bien dans le PATH avant /bin.
3. Installation du paquet utilitaire d'apparmor : Pour ce faire, il faut installer le paquet `apparmor-utils` (déjà présent dans la VM).

La documentation de l'outil se situe ici :
https://gitlab.com/apparmor/apparmor/wikis/Documentation

- Le mode "complain" sert à construire les profils.
- Le profil à rendre doit être en mode "enforce".
- La commande "ls" doit continuer de fonctionner sur l'ensemble du système excepter le dossier /boot.

Pensez à rendre le fichier *usr.local.bin.ls*.

1.11 Debian, RedHat et les autres distributions

Les différentes distribution GNU/Linux ont chacune leur façon d'approcher la sécurité. Par exemple, la distribution Debian propose dans des dépôts spécifiques l'usage de *AppArmor* pour protéger le noyau Linux, alors que la distribution CentOS va privilégier SElinux.

- Spécificités
  - Usage général vs spécifique (live, sécu, ...)
  - Orienté serveur vs client
  - Source vs binaire
  - Classique vs embarqué
  - *Rolling release* vs versions
- Historique : Trois grandes familles
  - Debian
  - Redhat
  - Slackware (base de Suse)
  - et plein d'autres . . . (distrowatch.com)

1.12 Exercice 3 : Contrôle d'intégrité et HIDS, un exemple avec AIDE

Les HIDS sont des systèmes de détection d'intrusion sur les hôtes. Pour ce faire, ils mettent en oeuvre, entre autre, une fonction de contrôle d'intégrité des éléments présents sur l'hôte ainsi surveillé.

Questions préliminaires

1. Citer 3 logiciels permettant d'effectuer du contrôle d'intégrité sur un hôte.
2. Citer une fonction autre que le contrôle d'intégrité que peut effectuer un HIDS.
3. Quel sont les éléments surveillé par logiciel de contrôle d'intégrité ?

Mise en pratique

Pour effectuer les taches et exercices de cette partie, le logiciel utilisé sera AIDE.

Pré-requis Installation du logiciel AIDE : Pour ce faire, il faut installer le paquet `aide`.

Documentation La documentation de l'outil se situe ici : `/usr/share/doc/aide-common/manual.html` et avec la commande `man aide.conf`

Partie 1 : Configurer les fichiers à vérifier La liste des fichiers à vérifier est :

- `/etc` : Intégrité de la donnée, des droits et owner/group, ctime et mtime.
- `/var/{lib,www,log}` : Intégrité des droits et des owner/group.
- `/{usr,}/{{s,}bin,lib*}` : Intégrité de la donnée, des droits et owner/group, ctime et mtime.
- `/root/*,/home/*` : Intégrité des owner/group.

Les sommes de contrôle utilisées sont md5 et sha256
Écrire le fichier de configuration `aide.conf` à utiliser.
Ce fichier est à rendre sous *ex3/aide.conf*

Partie 2 : Initialiser la base de donnée A l'aide de la commande appropriée, initialiser la base de donnée, et la placer sous `/var/lib/aide/aide.db`.
Ce ficher est à rendre sous *ex3/aide.db*

Partie 3 : Cron Écrire un script cron à placer sous `/etc/cron.d/aide`.
Celui-ci lancera aide pour faire une vérification du système toute les 10 minutes.
Pour le format des fichiers de type *crontab* : `man 5 crontab`
Ce fichier est à rendre sous *ex3/aide*

Questions subsidiaires

1. Que ce passe-t-il lorsque l'on modifie le fichier `/etc/resolv.conf` ?
2. Quels sont les modes d'avertissement que l'on peut avoir avec AIDE ?

1.13 Exercice 4 : Lynis

Lynis est un logiciel d'audit de sécurité qui permet de vérifier la conformité d'un système avec un modèle.

Pré-requis Installation du logiciel lynis : Pour ce faire, il faut installer le paquet `lynis` depuis les dépôts de lynis pour avoir la dernière version à jour :

`https://packages.cisofy.com/community/#debian-ubuntu`.

Partie 1 : Prise en main Prenez en main l'outil Lynis en analysant votre système avec le profil par défaut fournit par le paquet Debian.

Partie 2 : Création d'un profil *esiee* Modifier le profil par défaut pour avoir un profil *esiee* qui n'utilise que les tests suivants :

- mac_frameworks
- file_integrity
- authentication

Il faut donc sauter les autres tests.

Partie 3 : Ajout de vérification Ajouter un plugin esiee qui vérifie l'existence du fichier CRON de l'exercice sur AIDE.

Et ajoutez le au profile *esiee*.