Master 2 SeCReTS 2014-2015
Module Sécurité Système
Examen

Sécurité des systèmes Unix

Merci de bien lire les consignes :
- une seule feuille A4 manuscrite autorisée ;
- aucune communication ;
- aucun accès à un ordinateur, une station de travail, un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non ;
- sujet à remettre en fin d'examen.
- n'oubliez pas d'**indiquer nom et prénom sur la copie**.

Un point bonus est attribué au soin apporté à la rédaction des réponses.

Première partie

Sécurisation intrinsèque du système

Questions de cours (4pts)

1. *(0.5pt)* En quoi une négligence sur la sécurité physique d'un système peut réduire à néant tous les efforts faits sur la sécurité applicative de celui-ci ? Citez un exemple.
2. *(0.5pt)* Le terminal d'accès ultra-sécurisé d'une entreprise soucieuse de préserver ses brevets dans le domaine aéronaval demande à ses utilisateurs de s'authentifier grâce à un mot de passe de 16 caractères. Pour accéder au sous-réseau confidentiel, l'application demande en plus à l'utilisateur d'entrer un code PIN à 8 chiffres. S'agit-il d'une authentification dite « forte » ? Justifiez.
3. *(1.0pt)* Le mécanisme de mise en route du moteur d'une automobile futuriste est géré par un système embarqué basé sur une variante minimaliste d'Unix, qui implémente les PAM. Pour démarrer le véhicule, il est nécessaire que l'utilisateur soit sobre (vérification faite par `pam_ethylo.so`), et au choix : soit que l'utilisateur dispose d'une clef sans contact (`pam_nocontact.so`), soit qu'il connaisse le code PIN de démarrage (`pam_pinlock.so`). Rédigez la configuration du module.
4. *(0.5pt)* Voici un extrait de fichier `/etc/passwd` :

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
shutdown:x:500:500::/var/empty:/sbin/shutdown
romain:x:1000:1000:romain:/home/romain:/bin/bash
```

   Donnez le nom du compte captif, et expliquez brièvement à quoi il sert.
5. *(1.5pt)* Kevin souhaite mettre en place un *reverse-proxy* web sur son réseau personnel, et choisit d'installer un système dit « bastion ». Il opte pour la distribution Linux *Debian*. Il réalise un inventaire rapide des *packages* installés. Lesquels ne conviennent pas ? Pour chacun, expliquez pourquoi.

   - `acpid` : daemon de gestion des évènements de l'alimentation
   - `base-files` : divers fichiers de base du système Debian
   - `bash` : shell utilisateur
   - `bluez` : drivers et outils pour la gestion du bluetooth
   - `dpkg` : système de gestion des packages pour Debian
   - `gnuchess` : interface console pour le jeu d'échecs

— `initscripts` : scripts de lancement et d'arrêt du système
— `iptables` : outils d'administration pour le filtrage réseau
— `irssi` : client basé sur un terminal pour le protocole IRC
— `lvm2` : gestionnaire de volumes logiques Linux
— `mount` : outils pour monter les systèmes de fichiers
— `nmap` : simple scanner réseau
— `ntp` : daemon de synchronisation de l'horloge via le réseau
— `sox` : outils divers de manipulation et tranformation du son
— `ssh` : metapackage client et server pour le shell sécurisé
— `sudo` : outil de délégation de privilèges aux utilisateurs
— `tcpdump` : outil de surveillance et de capture réseau
— `tzdata` : données des fuseaux horaires
— `udev` : daemon de management des périphériques
— `xorg` : serveur graphique sous linux

Don't mess with myman (6pts)

L'objet de cet exercice porte sur l'exploitation d'un binaire nommé `myman`, un programme analogue à la commande `man` bien connue sur les systèmes Unix.

1. *(0.5pt)* Voici ce qu'on peut lire dans la console en essayant d'utiliser `myman` :

```
$ myman -h
usage: myman -h
       myman <fichier>
Affiche le contenu de /var/doc/<fichier> grâce à l'outil less.
$ myman ../../etc/debian_version
7.7
```

   Expliquez ce résultat. À quel type de faille est visiblement vulnérable cet outil ?

2. *(0.5pt)* Quel mécanisme mettriez-vous en place pour y remédier ?

3. *(1.0pt)* L'outil de mise en page est `less`.

```
$ ldd $(which less)
  linux-vdso.so.1 =>  (0x00007fff7e5ff000)
  libncurses.so.5 => /usr/lib/libncurses.so.5 (0x00007f77d51e4000)
  libc.so.6 => /lib/libc.so.6 (0x00007f77d4e82000)
  libdl.so.2 => /lib/libdl.so.2 (0x00007f77d4c7d000)
  /lib64/ld-linux-x86-64.so.2 (0x00007f77d5447000)
```

   Tracez l'arborescence du système de fichiers qu'il sera nécessaire de construire pour mettre en place le mécanisme évoqué à la question précédente. Expliquez la présence de chaque noeud de l'arbre que vous donnerez.

4. *(0.5pt)* Voici ce qu'on peut lire après un nouvel essai :

```
$ myman attention_ce_paramètre_est_le_plus_long_paramètre_du_monde_AAAA
Segmentation Fault
$ dmesg | grep segfault
[1337.314159] myman[4242]: segfault at 41414141 sp ffffd3a4 error 14
```

   Expliquez ce résultat. À quel type de faille est visiblement vulnérable cet outil ?

5. *(1.0pt)* Dessinez soigneusement la pile au moment du crash (sur un processeur x86). Vous devez faire apparaitre les éléments suivants, au minimum :
   — une flèche montrant le sens de croissance des adresses ;
   — une flèche montrant le sens de croissance de la pile ;
   — les sauvegardes diverses (adresse de retour, base de pile) ;
   — les paramètres des fonctions impliquées ;
   — les zones contenant des variables locales.

6. *(1.0pt)* Rappelez les deux conditions nécessaires pour qu'une vulnérabilité de ce type soit exploitable en pratique.

7. *(0.5pt)* Décrivez un type de payload que vous voudriez, en tant qu'attaquant, envoyer au programme vulnérable. Expliquez brièvement comment vous feriez.

Culture générale (1pt bonus)

Citez un évènement public en rapport avec la sécurité informatique paru ces derniers jours (conférence, publication, parution d'outil, d'exploit, de vulnérabilité, une attaque ciblée, etc.). Donnez quelques détails sur ce sujet en particulier. Si votre réponse est très satisfaisante, elle peut donner lieu à des points hors barème.

Deuxième partie

Prévention et confinement des attaques

1. *(1pt)* Sur la prévention des attaques :
   — Expliquez brièvement quel principe est utilisé pour protéger un programme contre les *buffer overflow* dans la pile au moment de la compilation.
2. *(2pts)* Un attaquant possède les informations suivantes sur le système qu'il cible : la pile des programmes est non-exécutable (NX), mais la bibliothèque C standard (`libc`) est toujours chargée à la même adresse.
   — Comment l'attaquant peut exploiter un *buffer overflow* dans la pile d'un programme vulnérable avec ces informations ?
   — Quelle fonctionnalité du noyau l'administrateur du système ciblé devrait activer pour éviter qu'une attaque réussisse ?
3. *(3pts)* Conteneurs et virtualisation
   — *(0.5pt)* Expliquez la différence fondamentale entre les mécanismes de virtualisation (comme VirtualBox) et les mécanismes de conteneurs (comme LXC ou Docker).
   — A l'aide d'un serveur peu puissant, on souhaite mettre en place une architecture de serveur Web contenant un *reverse-proxy*, un site web statique et un site web dynamique avec une base de données.
      — *(0.5pt)* Pour des raisons de sécurité, on souhaite séparer les environnements d'exécution des différents services. Quel mécanisme est le plus adapté pour cela ?
      — *(1pt)* Proposez un découpage de l'architecture en prenant en compte les différents services souhaités.
   — *(1pt)* On considère la sécurité du processus qui lit le certificat fournit pour établir les connexions SSL.
      — Ce processus est particulièrement vulnérable au moment où il lit le certificat. Expliquez pourquoi.
      — Que peut faire le processus pour limiter ses privilèges d'exécution au moment de la lecture du certificat ?
4. *(2.5pts)* Observez ce listing qui représente un profil AppArmor :

```
1  # Last Modified: Sun Sep 25 08:58:35 2011
2  #include <tunables/global>
3
4  /usr/sbin/rsyslogd {
5    #include <abstractions/base>
6    #include <abstractions/nameservice>
7
8    capability sys_tty_config,
9    capability dac_override,
10   capability dac_read_search,
11   capability setuid,
12   capability setgid,
13   capability sys_nice,
14   capability syslog,
15
16   # rsyslog configuration
17   /etc/rsyslog.conf r,
18   /etc/rsyslog.d/ r,
19   /etc/rsyslog.d/** r,
20   /{,var/}run/rsyslogd.pid rwk,
21   /var/spool/rsyslog/ r,
22   /var/spool/rsyslog/** rwk,
23
24   /usr/lib{,32,64}/rsyslog/*.so mr,
25
26   /dev/tty*                      rw,
27   /dev/xconsole                  rw,
28   @{PROC}/kmsg                   r,
```

```
29
30    /dev/log                        wl,
31    /var/lib/*/dev/log              wl,
32    /var/spool/postfix/dev/log      wl,
33
34    # 'r' is needed when using imfile
35    /var/log/**                     rw,
36
37    # Add these for mysql support
38    #/etc/mysql/my.cnf r,
39    #/{,var/}run/mysqld/mysqld.sock rw,
40
41    # Add thes for postgresql support
42    ##include <abstractions/openssl>
43    ##include <abstractions/ssl_certs>
44    #/{,var/}run/postgresql/.s.PGSQL.*[0-9] rw,
45
46    # Site-specific additions and overrides. See local/README for details.
47    #include <local/usr.sbin.rsyslogd>
48  }
```

— *(0.5pt)* Quelle est l'application concernée par ce profil AppArmor ?
— *(2pts)* Décrivez les règles présentes dans le profil par famille : réseau, accès aux fichiers, exécution de processus.

5. *(1.5pts)* On considère une installation standard de Apache, avec les éléments suivants :
   — racine du site web dans `/var/www` ;
   — configuration dans `/etc/apache2` ;
   — fichiers de log dans `/var/log/apache2` ;
   — modules dynamiques dans `/usr/lib/apache2`.
   Ecrivez un profil de configuration AppArmor correspondant à cette installation de Apache. Essayez de respecter le plus possible la syntaxe de AppArmor.

Question bonus *(1pt)* : Sur un système Linux configuré avec les fonctionnalités NX et ASLR, est-il intéressant d'utiliser un mécanisme comme AppArmor ? Donnez 2 arguments (pour ou contre).

Fin de l'examen.
