Module UE27 2009-2010
Sécurité des systèmes Unix

Mathieu BLANC
CEA/DAM

Consignes :
- 1h30 maximum ;
- tout document autorisé ;
- 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.

Deux points sont attribués au soin apporté à la rédaction des réponses.

---

1. *(4 points)* Que signifie la notion d'installation minimale d'une distribution Linux sur un ordinateur ? Quel est l'intérêt d'effectuer une installation minimale pour la securité ?

2. *(8 points)* Observez le code suivant :

```
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv) {
        char buf[16];
        printf("taille : %d\n", sizeof(buf));
        if ( argc < 2 )
                return -1;
        bzero(buf, sizeof(buf));
        strcpy(buf, argv[1]);
        printf("buf = %s\n", buf);
        return 0;
}
```

   (a) *(4 points)* Expliquez pourquoi ce programme, une fois compilé, présente un risque de débordement de tampon mémoire (*buffer overflow*) :
   - quelle est la taille déclarée du buffer ?
   - quel est la fonction qui risque de copier plus d'octets que permis par la taille du buffer ?

   (b) *(4 points)* Expliquez comment un attaquant peut faire pour communiquer une chaine de caractère au programme de façon à provoquer un crash.

3. *(6 points)* Observez le listing suivant :

```
% ldd /usr/sbin/lighttpd
        linux-gate.so.1 =>  (0x008c5000)
        libpcre.so.3 => /lib/libpcre.so.3 (0x007dc000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0x00dcf000)
        libattr.so.1 => /lib/libattr.so.1 (0x00365000)
        libssl.so.0.9.8 => /lib/i686/cmov/libssl.so.0.9.8 (0x00110000)
        libcrypto.so.0.9.8 => /lib/i686/cmov/libcrypto.so.0.9.8 (0x0015a000)
        libfam.so.0 => /usr/lib/libfam.so.0 (0x0062b000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0x00eaa000)
        /lib/ld-linux.so.2 (0x002d5000)
        libz.so.1 => /lib/libz.so.1 (0x0075f000)
        libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0x00bdc000)
```

   On souhaite faire fonctionner le programme `lighttpd` dans un environnement de type cage, aussi appelé `chroot()`.

   (a) *(3 points)* Que signifie le listing précédent ? Quel est le rapport avec la construction d'une arborescence spécifique pour la cage ?

   (b) *(3 points)* Quel est l'intérêt de contruire un environnement restreint pour le programme lighttpd ? Que doit-on faire en plus de construire une cage dans le système de fichier ?

Fin de l'examen.

Master 2 SeCReTS
Module UE27 2009-2010
Examen de rattrapage

Sécurité des systèmes Unix

Consignes :
- 2h maximum ;
- tout document autorisé ;
- 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.

Deux points sont attribués au soin apporté à la rédaction des réponses.

---

1. *(4 points)* Que signifie le sigle PAM ? Quel aspect de la sécurité des systèmes Unix est couvert par les PAM ? Les PAM constituent un mécanisme modulaire. Pourquoi ?
   Sur un système protégé par les PAM, on trouve le fichier `/etc/pam.d/login`. Que contient-il ?
2. *(8 points)* Observez le code suivant :

```
#include <stdio.h>
#include <string.h>

int main(int argc, char **argv) {
        char buf[16];
        printf("taille␣:␣%d\n", sizeof(buf));
        if ( argc < 2 )
                return -1;
        bzero(buf, sizeof(buf));
        strncpy(buf, argv[1], strlen(argv[1]));
        printf("buf␣=␣%s\n", buf);
        return 0;
}
```

   (a) *(4 points)* Expliquez pourquoi ce programme, une fois compilé, présente un risque de débordement de tampon mémoire (*buffer overflow*) :
   - quel est la fonction qui risque de copier plus d'octets que permis par la taille du buffer ? Pourquoi ?

   (b) *(4 points)* Comment un attaquant peut provoquer le débordement de buffer ?
   Donner un exemple de ligne de commande provoquant ce crash.

3. *(6 points)* Observez le listing suivant :

```
% ldd /usr/sbin/lighttpd
        linux-gate.so.1 =>  (0x008c5000)
        libpcre.so.3 => /lib/libpcre.so.3 (0x007dc000)
        libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0x00dcf000)
        libattr.so.1 => /lib/libattr.so.1 (0x00365000)
        libssl.so.0.9.8 => /lib/i686/cmov/libssl.so.0.9.8 (0x00110000)
        libcrypto.so.0.9.8 => /lib/i686/cmov/libcrypto.so.0.9.8 (0x0015a000)
        libfam.so.0 => /usr/lib/libfam.so.0 (0x0062b000)
        libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0x00eaa000)
        /lib/ld-linux.so.2 (0x002d5000)
        libz.so.1 => /lib/libz.so.1 (0x0075f000)
        libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0x00bdc000)
```

   On souhaite faire fonctionner le programme `lighttpd` dans un environnement de type cage, aussi appelé `chroot()`.

   (a) *(3 points)* Quel est l'intérêt de contruire un environnement restreint pour le programme lighttpd ?

   (b) *(1 points)* Comment prenez-vous en compte le listing pour la construction de l'environnement du programme `lighttpd` ?

   (c) *(2 points)* Citez deux autres mesures de précaution à appliquer à `lighttpd` pour minimiser ses privilèges d'exécution.

Fin de l'examen.