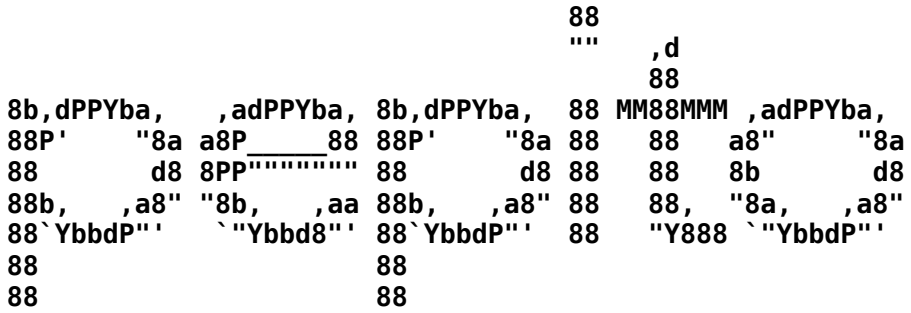

Correction des failles

Emilien Lancelot
Tristan Grellier
Alexandre Kalatzis

Buffer overflow

fonction strcpy(): L. 63[main.c]

Description: Si la variable **password** a une taille supérieure à la taille maximum de celle de **userPassword** alors on sort de l'espace alloué au programme.

main.c

```
- 63  strcpy(savePassword, password);
```

```
+ 63  if (strlen(password) > 512)
+ 64                {
+ 65                  printf("New password too long.\n");
+ 66                  return NOBODY;
+ 67                }
```

Explication de la correction: Si le nouveau mot de passe est supérieur à la taille maximum de **userPassword** alors aucune modification n'est faite et le programme affiche un message d'erreur: 'New password too long'.

L. 87[main.c]

Description: Si la variable **password** a une taille supérieure à la taille maximum de celle de **userPassword** alors on sort de l'espace alloué au programme.

main.c

```
- 87  if (password) {
- 88    strcpy(userPassword, password);
```

```
+ 87  if (password && my_strlen(password) < 512) {
+ 88    strcpy(userPassword, password);
```

Explication de la correction: Si le nouveau mot de passe est supérieur à la taille maximum de **userPassword** alors aucune modification n'est faite et le programme affiche un message d'erreur: 'New password too long'.

fonction strcpy() L.98[main.c]

Description: Si la variable **password** a une taille supérieure à la taille maximum de celle de **adminPassword** alors on sort de l'espace alloué au programme.

main.c

```
- 98  if (password) {
- 99          for (i = 0; password[i]; ++i)
-100               password[i] ^= xorKey;
-101          strcpy(adminPassword, password);
```

```
+ 98  if (password && my_strlen(password) < 512) {
+ 99          for (i = 0; password[i]; ++i)
+100             password[i] ^= xorKey;
+101          strcpy(adminPassword, password);
```

Explication de la correction: Si le nouveau mot de passe est supérieur à la taille maximum de **adminPassword** alors aucune modification n'est faite et le programme affiche un message d'erreur: 'New password too long'.

---

fonction strcat()

Description: Dans les trois fonctions strcat() qui permettent la composition de la variable **logMessage** . Si la chaine de caractère de droite est plus grande que celle de gauche il se produit alors un buffer overflow.

main.c

```
- 73  strcat(logMessage, "Invalid password : ");
- 74          strcat(logMessage, savePassword);
- 75          strcat(logMessage, "\n");
```

```
+ 73  strncat(logMessage, "Invalid password : ", strlen("Invalid password : "));
+ 74          strncat(logMessage, savePassword, strlen(savePassword));
+ 75          strncat(logMessage, "\n", strlen("\n"));
```

Explication de la correction: La fonction strncat() va couper la chaine à N caractères pour éviter un buffer overflow.

Integer overflow

fonction getNumber() L.303 [main.c]

Description: Aucune vérification n'est faite après l'appel à la fonction getNumber(). En effet si la variable corespondant au nombre d'ingredients demandés (amount) dépasse la taille d'un int il se produira certains phénomènes inattendus comme par exemple la variable qui passe dans le négatif ainsi qu'un gain d'argent incroyable.

main.c

```
-303  int   amount;
-304  amount = getNumber(&packetPtr, &packetSize);
-305  if ((money - 2 * amount) < 0) {
```

```
+303  int   amount;
+304  amount = getNumber(&packetPtr, &packetSize);
+305  if (amount < 0)
+306     amount = -amount;
+307  if (amount > 2147483647)
+308     {
+309         printf("You have reach the maximum amount of money.\n");
+310         return amount;
+312     }
+313  if ((money - 2 * amount) < 0) {
```

Explication de la correction: La première condition vérifie que la variable **amount** n'est pas inférieure à 0 et la seconde empêche la variable de dépasser la taille d'un int.

String format

fonction fprintf() L.74 [network.c]

Description: Cette utilisation de fprintf() pourait être une faille de sécurité potentielle. En effet, si le client envoie au server une chaine de caractères avec des flags de formatage, il peut alors détourner l'utilisation par défaut de la fonction.

main.c

```
- 74  fprintf(stderr, msg);
```

```
+ 74  fprintf(stderr, "%s\n", msg);
```

Explication de la correction: L'utilisateur ne peut plus envoyer de flags de formatage.

System, exécution de commandes externes

fonction system() L.317 [main.c]

Description: En modifiant la variable d'environement **$PATH**. Un utilisateur mal intentionné pourait rediriger l'appel à une fonction système par une de ses propres fonction.

main.c

```
-317  sprintf(log, "echo \"%s was bought\" >> log", ingredientName);
-318      free(ingredientName);
-319      system(log);
```

```
+317   if ((pid = fork()) < 0)
+318       return -1;
+319   if (pid == 0)
+320       {
+321           if ((ret = execl("echo", log)) == -1)
+322               printf("execl failed\n");
+323           exit(1);
+324       }
+325   else
+326       wait(&stay);
```

Explication de la correction: Les fonctions de type exec sont plus sécurisées car on ne peut en détourner l'utilisation.

Ld_preload

LD_PRELOAD est une variable d'environnement qui spécifie quelles bibliothèques partagées seront chargées avec les programmes lors de l'exécution. Lorsque cette variable est définie, l'éditeur de liens (Linker) chargera cette bibliothèque avant toutes les autres. Il devient alors possible de changer le fonctionnement de certaines fonctions.

Code:
-Cette ligne de code passe l'utilisateur en 'user':

```
if (!strcmp(password, userPassword))
       isUser = 1;
```

-Pour passer l'utilisateur en 'admin' il faut que cette instruction renvoie la valeur 0:

```
if (!strcmp(password, adminPassword))
    isAdmin = 1;
```

-Pour arriver à nos fins cette fonction va se substituer à la fonctin strcmp() du système grace au ld_preload.

```
int strcmp(const char *s1, const char *s2)
{
  static int i = 0;

  if (i == 0)
    {
      i = i + 1;
      return (1);
    }
  else
    return (0);
}
```

Explication de la correction: La première instruction ne sera pas prise en compte car elle renvera 1 alors que la seconde renverra bien 0. Le pepito pensera alors que nous possédons un compte administrateur et nous donnera accès à toutes les fonctionnalités du programme.

Autres failles de sécurité liées à l'utilisation de ld_preload

Il existe de nombreux moyens de détourner l'usage du programme. En transformant l'utilisation de strcpy() il est possible de copier une fausse chaine de caractères dans les variables globales **userPassword** ou **adminPassword**.

Obtention de contenu

Commande strings

Description: le binaire 'strings' permet d'obtenir toutes les chaines de caractères d'un binaire. On peut ainsi obtenir le mot de passe user du pepito, la version xor du mot de passe admin, le nom des fonctions utilisées, flags, etc...

Correction: L'obfuscation de code permet de rendre la lecture du résultat de la commande string beaucoup plus difficile. De plus, pour les mot de passe il n'est pas conseillé de les laisser en 'dur' dans le programme. Il est préférable de chiffrer tous les mots de passe en MD5, sha-1, sha-2, etc...
Les commentaires étant très explicites il est préférable de les supprimer.

Explication de la correction: Lorsque les commentaires étaient présents dans le binaire il pouvaient donner des indications sur l'utilisation du programme. Maintenant que ce n'est plus le cas, il est beaucoup plus compliqué d'obtenir la moindre information vraiment utile de la part de la commande string.

PTRACE:

fonction PTRACE:

Explication: Grace à un debugger comme gdb il est possible de modifier d'une façon simple le code assembleur directement. On peut ainsi par exemple inverser certaines conditions et donc changer le comportement du programme.

Correction du code:

```
if (ptrace(PTRACE_TRACEME, 0, NULL, NULL) < 0) {
    fprintf(stderr, "Debugger detected !\n");
    exit(EXIT_FAILURE);
    }
```

Explication de la correction: Si la fonction ptrace retourne la valeur -1 cela signifie que un debuger s'est attaché au processus courant. Si tel est le cas on quitte le programme.

Bugs

1) Entrer deux doubles quotes dans le read du client provoque le message d'erreur suivant: “list index out of range” puis l'arret soudain du client.
2) Changer le mot de passe existant par deux double quote puis demander à afficher les recettes provoque un segfault du serveur.