TP 1 : RISK

Par Aurélien Monnet-Paquet
M2 CySec

1. sudo apt-get install gcc-multilib
2. sudo apt-get install libc6-dev
3. sudo apt-get install libssl-dev:i386

CWE_131 (c)
Vulnérabilité :
CWE-131: Incorrect Calculation of Buffer Size
Le programme se base sur une entrée utilisateur sans vérification pour créer un buffer via un malloc.

Exploit :
Rentrer une valeur plus grande que MAX_INT.

Correction :
Effectuer une vérification de l'input ou changer le type de 'len' en un entier non signé.

CWE_131 (cpp)
Vulnérabilité :
Le mauvais calcul de la taille du buffer est au moment du malloc de SEQ_LEN. La taille d'un entier n'est pas pris en compte.

Correction :
Faire un malloc de SEQ_LEN * taille d'un entier.

CWE_134 (c)
Vulnérabilité :
Absence de gestion de formats de chaîne de caractères.
Confiance en l'utilisateur.

Exploit :
Avec une utilisation de '%s' et '%n' nous sommes capables d'écrire dans la variable miaou et ainsi changer la valeur.

Correction :
Ne jamais faire confiance à l'utilisateur.
Utiliser "%s" dans le printf et non pas argv[1] directement.

CWE_134 (cpp)
Vulnérabilité :
Idem que pour la version C.
La ligne : “printf(argv[1]);” n’est pas safe.
Même correction que pour la version C.

CWE_190 (c)
Vulnérabilité :
Integer Overflow.
Ici le programme effectue un calcul sur une entrée utilisateur et ne vérifie pas que le résultat soit correct.

Exploit :
Il suffit que n_resp soit supérieur à : (2**32 - 1) / sizeof(char*) et un Integer Overflow apparaît.

Correction :
Vérifier l’entrée utilisateur. Interrompre le programme si la valeur demandée est trop grande.

CWE_190 (cpp)
Vulnérabilité :
Integer Overflow dans le cadre d’une soustraction si ‘a’ est plus petit que ‘b’ dans l’opération suivante : “a - b” avec ‘a’ et ‘b’ des ‘uint32_t’.

Exploit :
Mettre le premier argument plus petit que le deuxième.

Correction :
Faire une vérification sur le fait que a doit être plus grand que b.
Changer les types de a et b en entier signé.

CWE_190 (Java)
Vulnérabilité :
Dans le cas ou l’utilisateur rentre un nombre plus grand qu’un ‘short’ alors il y a une boucle infinie. Puisque la limite d’un ‘short’ est de 65537 et cela restera toujours inférieur à la limite d’un entier.

Exploit :
Mettre en paramètre du programme n’importe quelle valeur plus grande que la limite d’un ‘short’.

Correction :
Tester le parametre pour verifier qu'il est dans la limite d'un 'short'.
Changer le type 'short' en 'int'.

CWE_798 (c)
Vulnérabilité :
Utilisation de login / mot de passe codés en dur dans le code de l'application.

Exploit :
Rien que la fonction "strings" de Linux affiche le login et mot de passe.

Correction :
Chiffrer les mots de passe avec une fonction cryptographique robuste.
Pas md5 ou sha-1 par exemple. sha-256 et sha-512 sont acceptables (avec un grain de sel).

CWE_798 (cpp)
Vulnérabilité :
Utilisation d'une mauvaise fonction de hachage et mot de passe pas salé.

Exploit :
Faire un "strings" du programme, sortir la chaîne md5 du mot de passe, et la rentrer dans une dictionnaire.

Correction :
Utilisé un grain de sel et une meilleure fonction de hachage.

CWE_798 (Java)
Idem que les précédents.

CWE_807 (c)
Vulnérabilité :
La confiance apportée à un tier dans une décision de sécurité.

Correction :
Ne faire confiance en personne.

CWE_807 (cpp)
Vulnérabilité :
Accord de privilèges sont authentification.

Correction :
Vérifier l'identité de root avant d'accorder les privilèges.