heig-vd
HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

SLO 2018

Sécurité Logicielle

Laboratoire 4

Exploitation de format strings

Département TIC - orientations TS/IL/IE

Professeur responsable :
Prof. Alexandre Duc

Assistants :
Loïc Haas
Lucie Steiner

25 mai 2018

1 Introduction

1.1 Déroulement et évaluation

Ce laboratoire sera évalué sur la base d'un rapport **individuel**, que vous devrez rendre d'ici au **mardi 12 juin à 23h50** sur CyberLearn. Ce rapport devra contenir les réponses aux questions (en vert), ainsi que **la description et le résultat** des manipulations (en bleu). **Tous les scripts** ainsi que le rapport devront être rendus dans une archive zip. Merci d'utiliser la même numérotation que dans ce document. Il n'est pas nécessaire de rédiger une introduction et une conclusion. De manière générale, préférez des réponses et des descriptions concises.

Tout le matériel nécessaire pour ce laboratoire est disponible sur CyberLearn et sur la page suivante :

`http://10.192.72.221/build`

Le serveur sur lequel se trouvent vos challenges est 10.192.72.221. Les ports sur lesquels vos challenges sont disponibles sont précisés dans le fichier "ports.pdf" qui se trouve sur CyberLearn. Les fichiers exécutables se trouvent sur la page http ://10.192.72.221/build, dans votre dossier personnel. Le code source de chaque challenge est à disposition sur CyberLearn.

Merci de rendre le rapport sous format **pdf**.

Conseils :

- Le premier exercice contient un mot complètement aléatoire. Recherchez-en la signification à vos risques et périls.
- Il peut être très utile de recompiler les programmes de son côté pour y ajouter des messages de debug et autre.
- Il est parfois nécessaire d'escaper avec un backslash les dollars.
- Pour écrire qu'un seul byte, il est possible d'utiliser le modificateur hh, par exemple : %6$hhn

2 Lecture

But : Récupérez le mot de passe secret et récupérez le flag.
Compilation : Le programme est compilé avec `gcc -Wno-format-security -no-pie -fno-PIC -m32 -O0`.
Vous avez le code source de ce programme mais pas le binaire.

Question 2.1

Quelle est la vulnérabilité dans ce code ? A quelle ligne se trouve cette vulnérabilité ?

Question 2.2

Comment pouvez-vous corriger ce code ?

Manipulation 2.1

Récupérez le mot de passe en remote et donnez le dans votre rapport.

Manipulation 2.2

Récupérez le flag et donnez le dans votre rapport.

3 Ecriture

But : Faire en sorte d'exécuter la fonction win et récupérer le flag.
Compilation : Le programme est compilé avec `gcc -Wno-format-security -no-pie -fno-PIC -m32 -O0`.
Vous avez le code source de ce programme ainsi que le binaire.

Question 3.1

Dans quelle partie de la mémoire se trouve le pointeur sur fonction ?

Question 3.2

Est-ce que l'ASLR est activé en remote ?

Manipulation 3.1

Affichez deux fois de suite l'adresse du pointeur sur fonction. Est-ce que cette adresse est modifiée ? Pourquoi ?

Question 3.3

Que doit-on faire pour exécuter la fonction win ?

Manipulation 3.2

Exploitez la vulnérabilité en remote pour accéder à la fonction win.

Question 3.4

Quel est le flag obtenu ?

Question 3.5

Est-ce que le fait d'avoir ou pas l'ASLR change l'attaque ? Pourquoi ?

4 Ecriture (2)

But : Faire en sorte d'exécuter la fonction win et récupérer le flag.
Compilation : Le programme est compilé avec `gcc -Wno-format-security -no-pie -fno-PIC -m32 -O0`.
Vous avez le code source de ce programme ainsi que le binaire.

Question 4.1

Est-ce que l'ASLR est activé en remote ?

Manipulation 4.1

Affichez deux fois de suite l'adresse du pointeur sur fonction. Que remarquez-vous ?

Question 4.2

Listez toutes les vulnérabilités du binaire. Comment peut-on les corriger ?

Question 4.3

Comment pouvez-vous récupérer l'adresse du pointeur sur fonction en remote ?

Manipulation 4.2

Sur la base de toutes ces informations, exploiter la vulnérabilité en remote pour exécuter la fonction win.

Question 4.4

Quel est le flag obtenu ?