Séance de TP : 13/11/2025

Plan de la séance

TP201 : Crackme en dynamique : échauffement

Ce petit programme a déjà été vu et vous le connaissez par cœur. Le but ici est de :

- Se familiariser avec GDB et les commandes de bases
- Se familiariser avec l'interface d'IDA
- Élaborer une stratégie de rétro-ingénierie

TP202 : Cas pratique de debug en dynamique : Etude du stackframe

- Manipuler GDB pour comprendre le stackframe sur un exemple simple

TP203 : Cas pratique de debug en dynamique : Etude des breakpoints softwares

- Manipuler GDB pour comprendre les breakpoints softwares

TP204 : Les structures de boucle et switch case

- IDA et GDB

TP205 : challenge

-

TP201 :

Partie 1 : ouvrir tp201 avec IDA :

- Trouver la fonction *strncmp* dans le programme et trouver l'adresse de l'appel.
- Lister les fonctions importées, afficher le CFG.
- Combien de basicblocks dans la fonction <main>, combien de chemins d'exécutions possibles ?
- Déduire le nombre de sauts conditionnels dans le programme
- A partir de la fonction strncmp() dans la section *.extern*, retrouver l'appel dans main avec la commande x. Quelle est le but de cette commande ?
- Comment IDA fait pour afficher les chaînes de caractères ?

Partie 2 : ouvrir tp201 avec GDB :

- Installer et ouvrir gdb avec le programme en argument
- Exécuter les commandes suivantes et interpréter les résultats :

```
(gdb) set disassembly-flavor intel
(gdb) maintenance info sections
(gdb) break main
(gdb) run
```

- Retrouver l'adresse de la section .text du programme dans IDA et GDB
- Expliquer les instructions suivantes :

```
(gdb) x/32i $rip
(gdb) ni
(gdb) si
(gdb) x/s $rax
(gdb) x/32xg $rsp
(gdb) pi
```

- Poser un breakpoint sur la fonction *strcmp*
- Expliquer exactement la création du stackframe
- Afficher les chaînes de caractère comparées
- Expliquer le fonctionnement du saut conditionnel au retour de strncmp. Observer les flags. Que pouvez-vous dire ?
- Conclure sur le crackme

TP202 : Stackframe

Ecrivez un programme qui fait apparaître une fonction et une sous fonction, puis utilisez GDB pour repondre aux questions suivantes :

- Quels sont les registres qui pilotent le stackframe ?
- A quoi sert-il ?
- Quelles sont les instructions responsables de sa création
- Quelles sont celles qui décide de la taille ?
- Que contient il ?
- Ajouter un buffer en variable locale, qu'observons-nous ?
- Décrire avec un dessin les dífférentes valeurs contenues dans le stack frame.

TP203 : Breakpoint softwaire

Ecrire un programme en C qui démontre le fonctionnement d'un breakpoint software.

Indices :

```
Utiliser le squelette suivant disponible tp203.c.
Poser un breakpoint sur une instruction et afficher le code binaire avant et après cette action.
```

TP204 : boucles et switch case

- Ecrire un code avec une boucle et observer le graphe dans IDA
- Retrouver la condition de fin de boucle, quelle est le compteur utilisé ?
- Refaire cet exercice avec le binaire de moodle tp204
- Idem pour le switchcase avec le tp204 aussi.

TP205 : Challenge

- En statique observer le CFG et décrivez-le (nb de basic block, de check, de chemins, boucle, switch etc..)
- Trouver une fonction ou une instruction sur laquelle poser un breakpoint
- Etablir une stratégie de résolution en dynamique. (On utilise les informations statiques pour établir une stratégie en dynamique)
- Poser un breakpoint sur la fonction main
- Suivez le paramètre de la fonction, affichez-le ainsi qu'une autre chaîne stockée dans le binaire.
- Quelles sont les transformations faites sur les paramètres ? Expliquez.