UQAC
Université du Québec
à Chicoutimi

6GEI186 - Architecture des ordinateurs

Travail 4 – Modification d'un fichier exécutable

à l'aide d'un désassembleur

Remis à : Prof. Jérémy Bouchard

Par:

David Chalons – CHAD17070000

Mobina Shamsadini – SHAM1352060

Samara Boudreault – BOUS08610400

Date de remise : 12 novembre

Objectif

L'objectif de ce laboratoire était d'analyser un exécutable avec x64dbg et d'en modifier le comportement lié à la vérification du mot de passe. Concrètement, il s'agissait de faire en sorte que le programme accepte n'importe quel mot de passe comme valide, puis de changer le texte affiché dans la fenêtre de confirmation. L'exercice visait à comprendre la logique d'un contrôle conditionnel en code machine, à manipuler des instructions d'assemblage simples et à réaliser des modifications binaires directes tout en préservant la stabilité du programme.

Partie 1 : Aperçu général (Modification du contrôle de mot de passe)

Nous avons observé que l'instruction originale « jne sample.59EA5A », dont l'opcode est « 75 00 », ne provoque en réalité aucun saut, car le déplacement indiqué est nul. Cela signifie que le programme poursuit simplement son exécution à l'instruction suivante sans modifier le flux normal du code.

Pour conserver ce comportement tout en simplifiant le code, nous avons remplacé cette instruction par « nop », abréviation de no operation dont l'opcode est « 66 90 ». Cette instruction indique au processeur de ne rien faire et de passer immédiatement à la suivante. Elle a donc le même effet que l'instruction d'origine, tout en étant plus explicite et plus propre.

Un autre avantage important est que « nop » occupe également deux octets, soit la même taille que « jne 00 ». Ce point est essentiel, car il permet de réaliser la modification sans risquer d'écraser les instructions voisines ni de décaler la structure du programme en mémoire.

Nous avons aussi envisagé d'autres solutions, comme l'instruction « cmovne rax, rax », qui produit un effet conditionnel équivalent. Cependant, cette dernière est plus longue (quatre octets) et aurait nécessité l'utilisation d'une zone de code libre, appelée code cave, rendant la modification inutilement complexe.

En conclusion, le remplacement de « jne » par « nop » constitue la solution la plus simple, la plus propre et la plus fiable. Cette approche préserve le comportement du programme tout en respectant la cohérence et la stabilité du code exécutable.

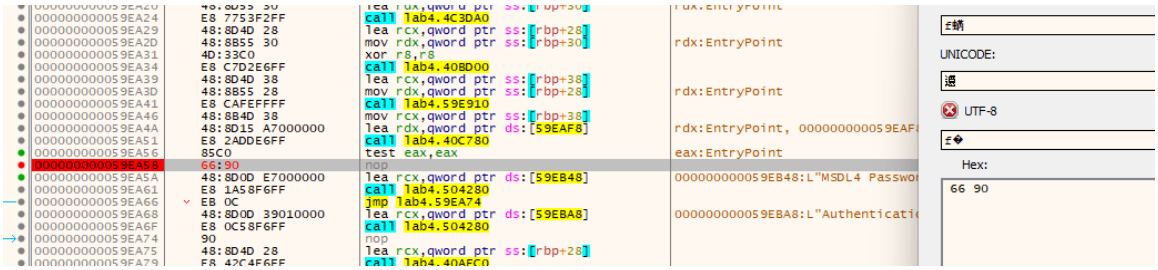

Figure 1 : Zone de code modifiée dans x64dbg montrant le remplacement de l'instruction jne(opcode 75 00) par nop (opcode 66 90)

Partie 2 : Modification de message affiché

En utilisant la fonction de recherche de chaînes de caractères dans x64dbg (Bouton AZ), nous avons localisé la zone mémoire qui contenait le message de confirmation affiché lorsque le mot de passe est accepté. Nous avons ensuite remplacé les octets ASCII correspondant au mot « Valid » par « MSDL4 », qui représente *Mobina, Samara, David - Lab 4* (les premières lettres de nos prénoms suivies de *L4*). Après cette modification, le programme affichait MSDL4 à la place de l'ancien message lorsque la vérification du mot de passe réussissait.

Vérification

On a relancé le programme et testé plusieurs mots de passe différents. Le programme suit toujours le chemin "valid" et affiche maintenant notre message personnalisé. La capture fournie montre la fenêtre après patch : *MSDL4 Password. Welcome, Authorized User.* (voir capture 2).

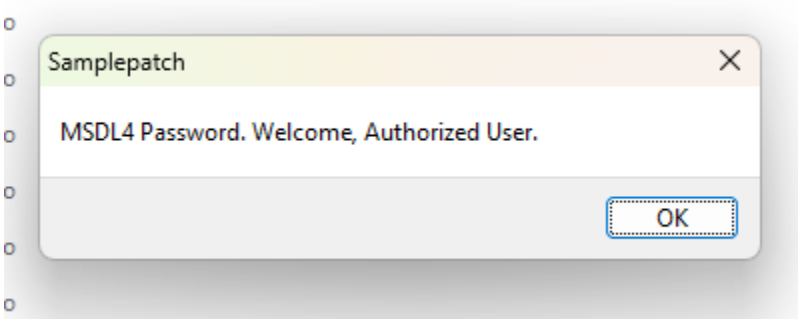

Figure 2:Fenêtre de confirmation affichée après la modification, indiquant le message personnalisé « MSDL4 Password. Welcome, Authorized User. »