Introduction au Game Hacking

Cet article présentera brièvement certains concepts des sciences de l'informatique et parlera des étapes nécessaires pour débuter dans le Game Hacking. Nous nous attarderons sur une technique fréquemment utilisée nommée Memory Editing et parlerons des outils nécessaires à la rétro ingénieurie d'un jeu vidéo.

Introduction au Game Hacking

Certaines personnes disent qu'un ordinateur n'est seulement qu'une série de 0 et de 1. Cela reviendrait à dire que le corps humain n'est qu'une série d'atomes. Cela n'est pas une information très utile. Les biologistes, les chimistes, parlent rarement d'atomes. Ils parlent plus souvent de Molécules. Cela ne veut pas dire que les atomes ne sont pas utiles, cela veut dire qu'ils ne sont pas toujours la meilleure manière d'aborder un problème. De manière similaire, pour traiter nos problèmes, il existe différents types de données similaires aux molécules, mais pour un ordinateur. Nous allons voir quel sont ces types de données et aborderont les bases du fonctionnement d'un ordinateur. Nous parlerons des diverses méthodes à apprendre pour hacker n'importe quel jeu. Puis pour finir, nous approfondirons une des méthodes la plus utilisée de nos jours: l'édition mémoire (Memory Editing). Cet article n'est qu'une introduction au Game Hacking. Il ne rentrera donc pas dans les innombrables détails qui existent dans ce domaine. Il a pour but de vous informer des techniques les plus connues et comprendre les démarches utilisées par les hackers. Pour prétendre être un véritable rétro-ingénieur et game hacker il faut compter de nombreuses années de recherches et d'apprentissage.

Les différentes compétences à développer:

Memory Editing
C'est l'une des compétences la plus utilisée pour modifier le comportement d'un jeu vidéo : modifier la vie de son personnage, le nombre d'argents, téléportation. Cette technique consiste à modifier des valeurs stockées dans la mémoire vive (RAM) de l'ordinateur. Cette mémoire est qualifiée de volatile. C'est-à-dire que lorsque le courant s'arrête, ne circule plus, ou que le système d'exploitation le juge utile, l'information stockée est libérée, perdue. Le système d'exploitation utilise cette mémoire pour enregistrer des informations temporaires. Nous en parlerons brièvement plus en détail plus tard.

Assembly Editing
C'est une compétence plus complexe et avancée qui vous permet de modifier des parties du code du jeu afin qu'elles réalisent ce que vous voulez. Cette compétence est plus compliquée à obtenir et nécessite de comprendre le langage Assembleur afin de pouvoir en lire et en écrire.

La différence avec l'édition mémoire peut être subtile. Par exemple, imaginons que sur votre jeu vous avez une barre de vie. Si vous souhaitez garder votre vie au maximum (100 points de vie) en utilisant le Memory Editing vous devrez utiliser une boucle qui s'exécutera toutes les millisecondes afin de remettre la valeur de la vie à son maximum (100) si celle-ci descend. Cela revient à "Freeze" la variable. Cela donne l'impression que votre vie ne descend pas, car la vitesse d'exécution de la boucle est extrêmement rapide. Cependant, un problème se pose, que se passe-

t-il si l'on prend des dégâts supérieurs à notre vie max ? Nous risquons de mourir, si cette boucle n'a pas eu le temps de remonter notre vie au maximum.

C'est pourquoi nous pouvons utiliser l'Assembly Editing. Plutôt que de continuellement remonter la vie de notre joueur dans une boucle nous allons désactiver le code qui lui fait perdre de la vie. Nous n'aurons donc plus de soucis avec la durée d'exécution de notre boucle et cela consommera sûrement moins de ressources sur notre ordinateur.

Hex Editing
C'est une compétence qui fait souvent illusion à l'édition des fichiers de sauvegarde d'un jeu, ou des fichiers présent sur le disque dur de l'ordinateur. C'est une technique qui est très peu utilisée de nos jours comparés au Memory Editing. Cependant, c'est une compétence à garder dans notre boîte à outils. Elle est notamment nécessaire pour modifier le comportement de jeux sur console, là où l'édition mémoire peut poser problème.

Packet Editing
Cette compétence est similaire à l'Hex Editing cependant au lieu d'éditer les fichiers de configuration du jeu vous allez éditer des paquets de données (unité de transmission utilisée pour communiquer au sein d'un réseau notamment internet) envoyés au serveur de jeu. Vous pouvez à ce moment faire croire au serveur de jeu que vous avez réalisé des actions qui n'ont jamais eu lieu.

Botting
Cela consiste à créer un programme qui joue à votre place. Souvent, cette compétence nécessite de maîtriser les précédentes afin d'être mise en place. Elle requiert de solides connaissances en programmation.

Il existe d'autres méthodes utilisées dans des cas plus spécifiques qui peuvent être utilisées pour hacker un jeu vidéo. En voici quelques-unes:

- Injection Graphique
- Resource Editing
- Game Hacking de jeu navigateur, (édition de JavaScript, Flash Decompiling)
- JVM/CLR Game Hacking
- Emulation de Jeu
- Contourner (ByPass) les systèmes Anti-Triche (Anti-cheat)
- Memory Editor Implémentation

Afin de pouvoir progresser, vous devez suivre un pattern d'apprentissage très simple lorsque vous voulez hacker un jeu vidéo. Cette démarche se base sur la méthode scientifique :

A) Trouvez ce que voulez modifier.
B) Émettez une hypothèse sur comment vous allez pouvoir modifier le jeu.
C) Mettez cette hypothèse en pratique et voyez si cela marche.
D) Si l'hypothèse émise n'a pas marché trouvez pourquoi et recommencez de nouveau en tenant compte de votre erreur.

En utilisant cette méthode très simple, vous apprendrez de vos erreurs et serez en mesure d'hacker n'importe quel jeu.

Zoom sur l'édition mémoire (Memory Editing)

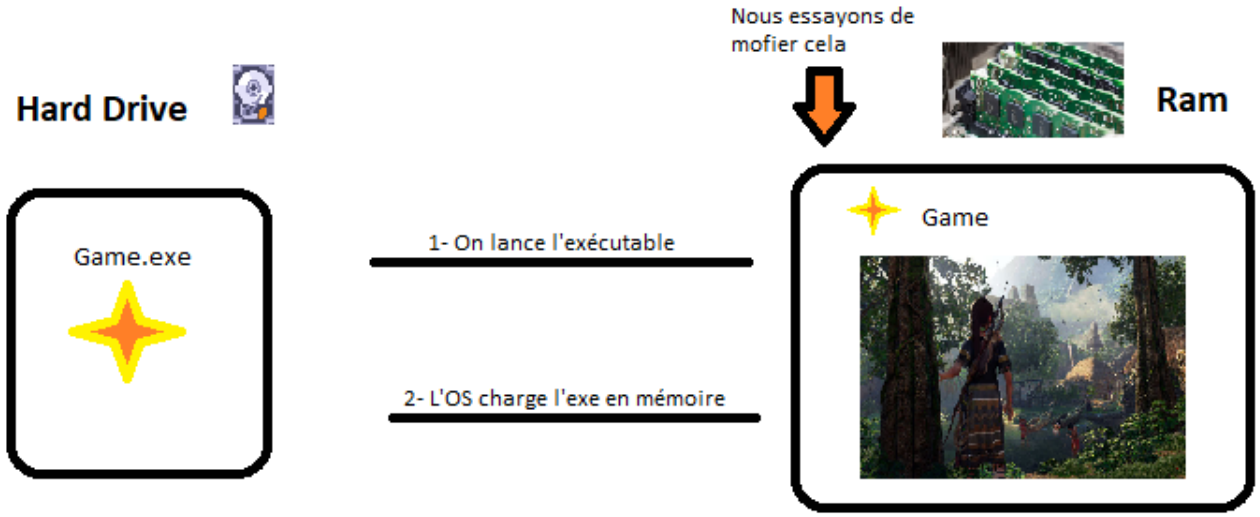
Un éditeur de mémoire aide les hackers à comprendre le système d'exploitation et permet d'effectuer une analyse de la mémoire. Apprendre à pirater/modifier la mémoire est une base dans le hacking de jeu vidéo. De nombreux hackers à travers le monde, tels que ceux travaillant pour certains États, les black hat, les professionnels des tests d'intrusion (pentesters), savent comment utiliser ces concepts pour effectuer des piratages parmi les plus avancés et les plus importants dans le monde.

Parlons un peu du système d'exploitation (OS):

Le système d'exploitation gère de nombreuses choses, parmi les plus importantes:

- Il permet de lier le hardware et le software, c'est-à-dire qu'il permet de coordoner les composants matériels pour faire fonctionner des logiciels.
- Il permet de gérer les différents processus s'exécutant sur l'ordinateur selon un ordre d'ordonancement.
- Il permet la gestion de la mémoire.
- Il permet la gestion des périphériques d'Entrées et Sorties.
- Il permet la gestion du système de fichier.

Lorsque vous lancez un exécutable, le système d'exploitation le charge en mémoire, c'est cela que nous essayons de modifier. Bien évidemment, de nombreuses choses se passe également mais nous n'en avons pas besoin pour le moment. Comme nous modifions ce qui se trouve en mémoire (RAM). Les changements n'affectent pas le disque, ils ne sont donc pas permanents.



Sachant que le système d'exploitation est responsable de la gestion des processus, nous pouvons abuser de ses droits pour modifier notre processus cible. Ici, notre jeu. Chaque système d'exploitation possède une API, soit des fonctions que nous pouvons utiliser dans notre code pour

développer nos cheats. Certaines de ces fonctions sont très puissantes. Certaines nous permettent d'éditer ce qui se trouve en mémoire RAM.

Voir doc: https://docs.microsoft.com/en-us/windows/win32/api/memoryapi/nf-memoryapi-writeprocessmemory

Si nous souhaitons modifier des valeurs dans notre jeu, comme notre position, nous allons **scanner** la mémoire du processus correspondant au jeu à la recherche de ces valeurs. Lorsque nous trouvons le nombre que nous voulons changer nous allons alors **éditer** cette mémoire par le nombre que nous souhaitons.

Memory Scanning:

Mais tout d'abord qu'est ce qu'une adresse mémoire ?

Une adresse mémoire est un nombre entier naturel qui désigne une zone particulière de la mémoire, ou juste le début d'une zone. Imaginez que la mémoire RAM correspond au quartier de votre amie, sa maison correspondra à l'adresse mémoire, et votre amie la valeur recherchée, ici le nombre de points de vie 100.
Lorsque nous scannons la mémoire par exemple à la recherche de nos points de vie (100) nous cherchons à trouver l'adresse ou est stocké le nombre 100. Cela revient à dire que nous sommes à la recherche de votre amie dans son quartier et cherchons à connaître l'adresse de sa maison afin de la retrouver plus facilement par la suite.

Pour modifier notre nombre de points de vie, il suffira d'utiliser notre logiciel pour éditer la mémoire. Nous utiliserons l'adresse mémoire pour retrouver nos points de vie, et modifier leur valeur.

Il existe un logiciel très utile et souvent utilisé par les Game Hacker, il s'appelle Cheat Engine. Il permet à la fois de scanner la mémoire du jeu et de l'éditer.

Pour apprendre à maîtriser le logiciel, je vous invite à suivre la série de vidéos sur Cheat Engine de Guided Hacking.

Comme vous le remarquerez lorsque vous allez redémarrer votre jeu les adresses mémoires auront sûrement changé, comment pouvons nous alors modifier la valeur de nos points de vie ? Comment peut ont trouver la position de notre joueur si nous souhaitons écrire un hack de téléportation, mais n'avons aucune idée de sa valeur ? Je vous invite à poursuivre par vous-même en approfondissant les diverses thématiques abordées. Il existe également un très bon forum pour débuter dans le game hacking. Toutes les bases nécessaires et sujets y sont enseignés : Cliquer Ici --> Guided Hacking.