Aufgabe 03

Simon Kramer

May 18, 2023

1 Aufgabe

Es wurde ein Programm namens CrackMe.exe gegeben. Dieses erwartet ein String im Format von "xxxx-xxxx-xxxx-xxxx". Wir sollen durch Reverse Engineering herausfinden, wie dieser Code aufgebaut sein muss, damit man das Produkt aktivieren kann. Dafür mussten wir x32dbg an den Prozess anhängen und das Programm analysieren.

2 Analyse

Zuerst wird beim Anheften des Debuggers das Programm beim Starten pausiert. Danach kann man nach unterschiedlichen Strings suchen. So konnten die Stings, die aussagen, dass das Programm aktiviert wurde oder dass die Aktivierung fehlgeschlagen ist, gefunden werden. Kurz vor der Ausgabe der Strings, befindet sich der Codeblock, der die Validität des Strings überprüft.

Die erste Zahl wird +16 und +0xFFFFFAD7 gerechnet. Da die gesamte Zahl nur 0xFFFFFFFF sein kann führt alles über 538 zu einem Overflow. Das muss genutzt werden, weil die Zahl nach der Addition kleiner als 2000 sein muss. Die gesamt Range ist 1337 - 3337

Die zweite Zahl wird +0xFFFFE357 gerechnet. Da die gesamte Zahl nur 0xFFFFFFFF sein kann führt alles über 0x1CA8 (dez:7337) zu einem Overflow. Das muss genutzt werden, weil die Zahl nach der Addition kleiner als 2000 sein muss. Die gesamt Range ist 7337 - 9337.

Die dritte Zahl muss nach 0x03E8 größer sein als 999, also mindestens 1000. Zudem wird das least significant byte überprüft. Diese muss gleich eins sein. Die Zahl muss also ungerade sein.

Die vierte Zahl muss nach 0x03E8 größer sein als 999. Zudem wird das least significant byte überprüft. Diese muss ungleich eins sein. Die Zahl muss also gerade sein.

Ein möglicher Key ist: "1337-7337-1001-1000"

3 Pseudo Code

```
CheckSum(int[] input):bool{
        if input.Length != 4    : throw exception;

        if input[0] < 1337      : return false
        if input[0] > 3337      : return false

        if input[1] < 7337      : return false
        if input[1] > 9337      : return false

        if input[2] < 1000      : return false
        if input[2] % 2 != 1    : return false

        if input[3] < 1000      : return false
        if input[3] % 2 == 1    : return false
        return true;
}
```