Igor Chorążewicz 160733

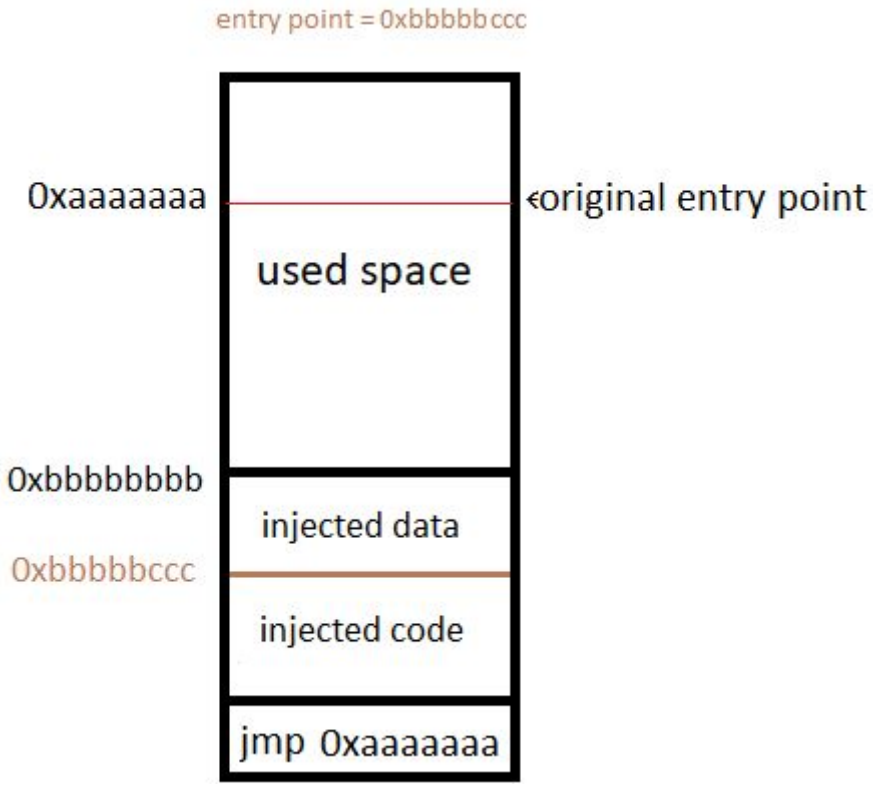
# Modyfikacja pliku PE

## Cel

Celem projektu było wstrzyknięcie własnego kodu do pliku wykonywalnego PE. Jako rezultat powstał program pozwalający na zmodyfikowanie pliku PE w taki sposób, że po jego uruchomieniu zostanie wywołana funkcja WinAPI (np. MessageBox) a następnie sterowanie powróci do oryginalnego programu.

## Sposób wykonania

Główna idea polega na skopiowaniu rozkazów (kodu, który chcemy wstrzyknąć) w postaci binarnej na koniec segmentu kodu w docelowym programie. Należy także dodać instrukcję skoku pod adres oryginalnego punktu wejściowego programu (entry point), oraz zmodyfikować ten adres, tak aby wskazywał na adres początku naszego doklejonego kodu.




Wstrzykiwany kod jest kompilowany w ramach programu wstrzykującego - rozkazy w formie binarnej są kopiowane do odpowiedniego miejsca w modyfikowanym pliku. Po skopiowaniu ustalane są dokładne wartości adresów funkcji (patrz opis niżej). Dla uproszczenia dane (nazwy funkcji) także trzymane są w segmencie kodu, są one

umieszczane na samym początku doklejanego obszaru. Adres entry point jest ustawiany na pierwszą instrukcję (z pominięciem danych).

Celem wstrzykiwanego kodu jest wywołanie funkcji z WinAPI, tak więc należy posłużyć się funkcją *LoadLibrary* oraz *GetProcAddress* aby załadować odpowiednią bibliotekę i otrzymać adres szukanej funkcji. Najłatwiejszym sposobem wywołania dwóch wyżej wymienionych funkcji jest wykonanie instrukcji *call* podając jako argument adres wywoływanej funkcji (można go odczytać z tablicy eksportów w pliku kernel32.dll). Problemem jest jednak zmienność tych adresów (np. po aktualizacji systemu, mechanizm randomizacji adresów), dlatego lepszym pomysłem jest każdorazowe wyznaczenie tego adresu. Jest to realizowane w następujący sposób: wstrzykiwany kod początkowo nie zawiera adresów powyższych funkcji a jedynie specjalne wartości (magic numbers) - np. 0x12341234 dla *LoadLibrary* i 0x11223344 dla *GetProcAddress.* Po ustaleniu adresu bazowego biblioteki kernel32.dll powyższe wartości są zastępowane prawidłowymi adresami (suma adresu bazowego biblioteki oraz offsetu funkcji w tej bibliotece). Można to zrobić korzystając z bloku TEB (Thread Environment Block), który zawiera informacje o adresach pod który załadowane zostały biblioteki dynamiczne. Do bloku TEB można dostać się korzystając z rejestru segmentowego FS - jego adres to **fs:18h.**

## Dodatkowe informacje

Program realizujący opisaną funkcjonalność nie będzie działał dla wszystkich programów. W szczególności dla programów systemowych windows 10 wstrzyknięcie kodu sprawia, że docelowy program nie uruchamia się. Pod debuggerem można się przekonać, że program zawiesza się na instrukcji *int 29h*. Prawdopodobnie jest to skutek działania mechanizmu Control Flow Guard.

Program, który wstrzykuje kod powinien być kompilowany w wersji *Release* lub z wyłączoną opcją *incremental linking.* W przeciwnym razie kompilator będzie stosował dwustopniowe wywoływanie funkcji, w konsekwencji jeśli w języku C pobierzemy adres funkcji i będziemy chcieli odczytać rozkazy jakie ta funkcja zawiera, nie uda nam się to. Adres ten będzie wskazywał na obszar pamięci w którym wystąpi instrukcja jmp xxxx, gdzie xxxx jest faktycznym adresem funkcji w pamięci (adresem rozkazów funkcji).

Kod: https://github.com/chorig9/pe_modify