Marcin Bukowiecki

Nr indeksu: 209799

Grupa środa 15:15

1. Zadaniem na laboratorium 5, było zapoznanie się z kodem assemblera, który został zaimplementowany w programie „patch_me_sr”. Następnie dostosowanie kodu programu tak aby przyjmowane hasło zwracała komunikat sukcesu.
2. Analiza kodu opierała się na zrozumienie funkcji „main”.
   Program wywoływał w takiej kolejności funkcje:
   Funkcja wypełniała tablicę wywołaniem funkcji looser a tylko pod jednym indeksem, który był wyliczony na podstawie hasła zostało umieszczone wywołanie funkcji winner.
   call 804857a <init>
   Funkcja time_guar() działa w następującej logice:

```
int time_guard(){
        if(time()-g1<3){
                g1=time();
        result = 0;
        }else return -1
return result;
}
```

   Wyświetlenie komunikatu o podanie hasła
   call 80483a8 <printf@plt>
   odczyt hasła
   call 80483c8 <__isoc99_scanf@plt>
   Następnie w main wyliczany jest indeks do tablicy z wywoływaniem funkcji winner bądź looser na postawie podanego hasła
   call *%eax

   Aby program akceptował hasło było kilka możliwości:
   - ostatnie wywołanie w main ustawić na wywołanie funkcji winner w ten sposób zawsze otrzymamy pozytywny wynik.
   Ten sposób zmuszał do ustawienia na jednym bajcie adresu funkcji winner a na kolejnym nop (0x90).
   - kolejnym sposobem było wypełnienie tablicy na wywołania funkcji wszystkich elementów na winner

   Do rozwiązania powyższych problemów należało zmodyfikować plik binarny programu.
   Do tego celu użyliśmy programu vim w trybie heksadecymalnym (%!xxd),następnie należało w odnalezionym fragmencie adresu ustawić odpowiednie bajty.

   Pierwszy problem: w liniach 804858c, 80485a7 ustawiono taki sam ciąg bajtów.

```
804858c:        ba 66 85 04 08          mov   $0x8048566,%edx
80485a7:        ba 66 85 04 08          mov   $0x8048566,%edx
```

   Drugi problem: w linii 8048661 ustawiono wywołanie funkcji winner reszt uzupełniono nop.

```
8048661:        66 90           call  *%eax
```

Należało również w funkcji time_guard w sekcji zwracania -1 zmienić na 0:
Linia 80484db: ustawiono wartość -> b8 00 00 00 00.
Niestety te sposoby wymagały uwzględnienie funkcji time_guard, wystarczyło ustawić na wyjściu funkcji zawsze zero. Ten sposób uchronił nas przed ograniczeniem czasowym.
Te dwa sposób zostały wykonane podczas zajęć laboratoryjnych.
W ten sposób wprowadzenie, każdego hasła dawało wynik pozytywny.

3. Wnioski:
   Zajęcia pomogły zrozumieć, że na kilka sposobów można znaleźć rozwiązania w programie aby złamać hasło. Podczas zajęć mieliśmy okazję przypomnieć sobie kilka komend assemblera oraz krokowo rozwiązać zadanie, którym celem było uzyskać wynik pozytywny programu.