SPAASM - Systémové programovanie a assemblery

Zadanie č.3 - Statická a dynamická analýza programu

Autor: Marek Čederle

Vypracovanie

Na vypracovanie úloh som použil nástroje:

- `HxD` : Hex Editor
- `Ghidra` : Reverse Engineering Framework

1. Správna dĺžka akceptovaného reťazca je 8 znakov. Pri disassemblingu som zistil že funkcia ktorá načítava zadávaný reťazec ukladá svoju návratovú hodnotu do premennej ktorá sa potom porovnáva s hodnotou 8 a vtedy sa vykonávajú ďalšie časti programu. Táto hodnota sa opakuje vo viacerých nasledujúsich častiach programu čo ale spomeniem v ďalších úlohách.
2. Správny reťazec je `FIITgeek`. Program funguje nasledovne:
   - Program pomocou funkcie `GetDlgItemTextA` načíta reťazec a uloží ho na adresu `0x00403058`, pričom daná funkcia pri úspechu vráti počet znakov reťazca. Ak je táto hodnota iná od 8 tak sa nastaví výstupná hláška na `Wrong !` a zobrazí sa okno s touto hláškou. Ak je hodnota 8, tak sa zavolá funkcia ktorej parametre sú dva reťazce (`I4561AsEmblerySuPOhodicka2x3XzgvwpqLJfBCDnFMH90KNG78QRtTUVWjYZ` , `J#ki80Ys`) pričom úlohou danej funkcie je skopírovať znaky z prvého reťazca do druhého reťazca na určené pozície. Takto sa tam skopíruje reťazec `FIITgeek` znak po znaku. Táto funkcia síce nepriradzuje svoju návratovú hodnotu žiadnej premennej ale vráti adresu reťazca `J#ki80Ys` (teraz už `FIITgeek`), ktorá sa uloží do registri `EAX`. Táto hodnota sa následne skopíruje z registra `EAX` do `EBX`. Následne sa zavolá ďalšia funkcia, ktorej parametre sú adresa načítaného reťazca a dĺžka očakávaného reťazca. Táto funkcia následne porovnáva znak po znaku reťazec v `EBX` a reťazec na adrese `0x00403058` a ak sa zhodujú tak sa pomocou návratovej hodnoty nastaví register `EAX` na `0x1`. Tento register sa následne porovnáva s konštantou `0x1`, pričom ak sa rovnajú, tak sa nastaví výstupná hláška na `Right !` a zobrazí sa okno s touto hláškou. Ak sa reťazce nezhodujú tak sa nastaví výstupná hláška na `Wrong !` a zobrazí sa okno s touto hláškou.
3. - `DialogBoxParam`
     - Argumenty:

            ```
            HINSTANCE hInstance,
            LPCSTR    lpTemplateName,
            HWND      hWndParent,
            DLGPROC   lpDialogFunc,
            LPARAM    dwInitParam
            ```
        - Návratová hodnota: `INT_PTR`, ak funkcia uspeje, návratovou hodnotou je hodnota parametra `nResult` uvedená vo volaní funkcie EndDialog použitej na ukončenie dialógového okna. Ak funkcia zlyhá, pretože parameter `hWndParent` je neplatný, návratová hodnota je nula. Funkcia v tomto prípade vracia nulu kvôli kompatibilite s predchádzajúcimi verziami systému Windows. Ak funkcia zlyhá z akéhokoľvek iného dôvodu, návratová hodnota je -1.
        - Popis: Vytvorí dialógové okno zo šablóny dialógového okna. Pred zobrazením dialógového okna funkcia odovzdá procedúre dialógového okna hodnotu definovanú aplikáciou ako parameter `lParam` správy `WM_INITDIALOG`. Aplikácia môže túto hodnotu použiť na inicializáciu ovládacích prvkov dialógového okna.
    - `GetDlgItemText`
        - Argumenty:
            ```
            HWND  hDlg,
            int   nIDDlgItem,
            LPSTR lpString,
            int   cchMax
            ```
        - Návratová hodnota: Ak funkcia uspeje, návratová hodnota udáva počet znakov skopírovaných do vyrovnávacej pamäte bez koncového nulového znaku. Ak funkcia zlyhá, návratová hodnota je nula.
        - Popis: Získa názov alebo text priradený k ovládaciemu prvku v dialógovom okne.
    - `MessageBoxA`
        - Argumenty:
            ```
            HWND    hWnd,
            LPCTSTR lpText,
            LPCTSTR lpCaption,
            UINT    uType
            ```
        - Návratová hodnota: Vracia celočíselnú hodnotu, ktorá označuje, na ktoré tlačidlo používateľ klikol. Ak funkcia zlyhá, návratová hodnota je nula.
        - Popis: Zobrazí dialógové okno, ktoré obsahuje systémovú ikonu, sadu tlačidiel a krátku správu špecifickú pre aplikáciu, napríklad informácie o stave alebo chybe.
4. `GetDlgItemText` sa volá na adrese `0x00401097` a získa text, ktorý bol napísaný v dialógovom okne.

5. `DialogBoxParam` sa volá na adrese `0x0040101e` a vytvorí dialógové okno zo šablóny dialógového okna.
6. `MessageBox` sa volá na adrese `0x004010ea` a pri správnom zadaní reťazca sa zobrazí hláška `Right !`.
7. Správny reťazec je teraz `Marek`. Nižšie je napísaný postup ktorým som to dosiahol a čo som zmenil.

Najskôr som skúšal to napasovať na reťazec `marekced` s inštrukciami tak, že pomením iba offsety, ale to sa mi nedarilo pretože mi nesedeli dĺžky jednotlivých inštrukcií, tak som potom zvolil cestu že skrátim retazec na `Marek` napíšem inštrukcie ako potrebujem a zvyšok čo mi zostane (aby sedeli počty a adresy inštrukcí) doplním inštrukciami `NOP`.

Reťazec `I4561AsEmblerySuPOhodicka2x3XzgvwpqLJfBCDnFMH90KNG78QRtTUVWjYZ`:

- dĺžka 62
- variable[offset] if (offset = 62) (0x3e) sa nachádza null char, ktorý tiež potrebujem skopírovať

OFFSETy vo veľkom reťazci

char	dec	hex
M	43	0x2b
a	24	0x18
r	12	0xc
e	11	0xb
k	23	0x17
\0	62	0x3e

- Zmenené offsety v kopírovacej funkcii a potom pridané NOPy, aby sedeli inštrukcie
- Na 3 miestach zmenené miesto čísla 8 číslo 5 ako veľkosť reťazca:
  - návratová hodnota funkcie, kde sa deje kopírovanie reťazca
  - if podmienka v hlavnej funkcii
  - argument v porovnávacej funkcii

Kopírovacia funkcia + návratová hodnota

Pôvodné	Nové
00401146 PUSH EBP	00401146 PUSH EBP
00401147 MOV EBP ,ESP	00401147 MOV EBP,ESP
00401149 PUSH ESI	00401149 PUSH ESI
0040114a PUSH EDI	0040114a PUSH EDI
0040114b XOR EAX ,EAX	0040114b XOR EAX,EAX
0040114d MOV ESI ,dword ptr [EBP + param_1]	0040114d MOV ESI,dword ptr [EBP + param_1]
00401150 MOV EDI ,dword ptr [EBP + param_2]	00401150 MOV EDI,dword ptr [EBP + param_2]
00401153 MOV AL ,byte ptr [ESI + 0x2a]	00401153 MOV AL,byte ptr [ESI + 0x2b]
00401156 MOV byte ptr [EDI],AL	00401156 MOV byte ptr [EDI],AL
00401158 MOV AL ,byte ptr [ESI]	00401158 MOV AL,byte ptr [ESI + 0x18]
0040115a MOV byte ptr [EDI + 0x1],AL	0040115b MOV byte ptr [EDI + 0x1],AL
0040115d MOV AL ,byte ptr [ESI]	0040115e MOV AL,byte ptr [ESI + 0xc]
00401160 MOV byte ptr [EDI + 0x2],AL	00401161 MOV byte ptr [EDI + 0x2],AL
00401163 MOV AL ,byte ptr [ESI + 0x37]	00401164 MOV AL,byte ptr [ESI + 0xb]
00401166 MOV byte ptr [EDI + 0x3],AL	00401167 MOV byte ptr [EDI + 0x3],AL
00401169 MOV AL ,byte ptr [ESI + 0x1e]	0040116a MOV AL,byte ptr [ESI + 0x17]
0040116c MOV byte ptr [EDI + 0x4],AL	0040116d MOV byte ptr [EDI + 0x4],AL
0040116f MOV AL ,byte ptr [ESI + 0xb]	00401170 MOV AL,byte ptr [ESI + 0x3e]
00401172 MOV byte ptr [EDI + 0x5],AL	00401173 MOV byte ptr [EDI + 0x5],AL
00401175 MOV AL ,byte ptr [ESI + 0xb]	00401177 NOP
00401178 MOV byte ptr [EDI + 0x6],AL	... NOP
0040117b MOV AL ,byte ptr [ESI + 0x17]	0040117f NOP
0040117e MOV byte ptr [EDI + 0x7],AL	00401180 NOP
00401181 MOV EAX ,EDI	00401181 MOV EAX,EDI
00401183 POP EDI	00401183 POP EDI
00401184 POP ESI	00401184 POP ESI

Pôvodné	Nové
00401185 LEAVE	00401185 LEAVE
00401186 RET 0x8	00401186 RET 0x5

if podmienka v hlavnej funkcii

Pôvodné	Nové
0040109d CMP EAX,0x8	0040109d CMP EAX,0x5

argument v porovnávacej funkcii

Pôvodné	Nové
004010ba MOV EAX,0x8	004010ba MOV EAX,0x5

Zdroje

- Ghidra
- DialogBoxParamA
- GetDlgItemTextA
- MessageBoxA