

Reverse Engineering Lab 5

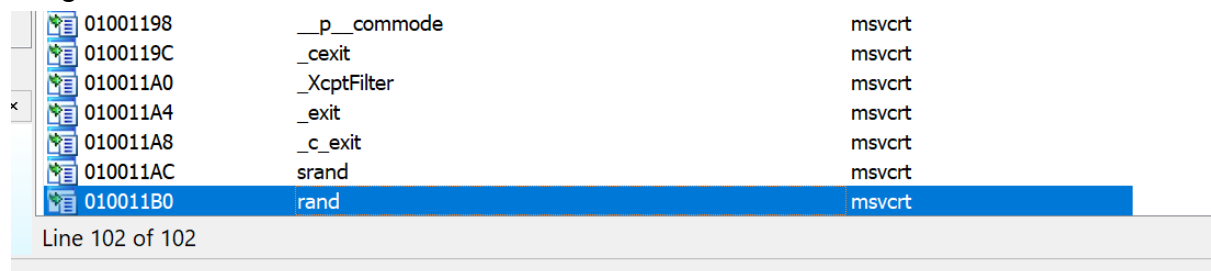
Minesweeper

Radu Dilirici, 510

1. Identificarea Bombelor

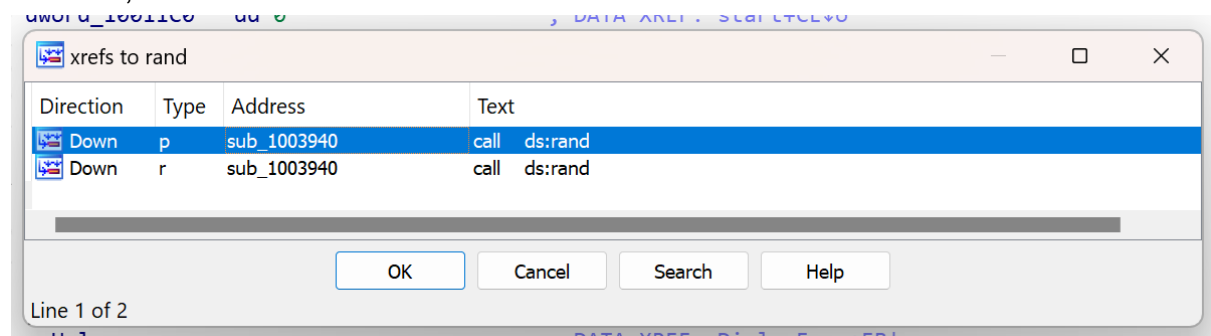
Initial am presupus ca jocul se initializeaza la apasarea primei casute, astfel incat sa nu poti apasa prima oara pe o bomba. De aceea, am analizat in x32dbg (varianta 32 bits a lui x64dbg) ce se executa la apasarea primei casute, insa nu am avut succes. Se pare ca jocul este initializat de la bun inceput, iar daca cumva jucatorul apasa din prima pe o bomba, programul o muta in alta parte.

Pentru ca generarea campului ar trebui sa se intample intr-un mod aleator (alta presupunere) am cautat in sectiunea de imports din IDA o functie de random si am gasit **rand**.



```
.idata:010011AC ; DATA XREF: sub_1003AB0+E↓r
.idata:010011B0 ; int __cdecl rand()
.idata:010011B0 extrn rand:dword ; CODE XREF: sub_1003940↓p
.idata:010011B0 ; DATA XREF: sub_1003940↓r
.idata:010011B4
.idata:010011B4
```

Apoi m-am uitat in ce locuri este folosita functia **rand**. Aceasta avea doua referinte, iar ambele in aceeaasi functie.



```
1 int __stdcall sub_1003940(int a1)
2 {
3     return rand() % a1;
4 }
```

Din “traducerea” functiei in pseudocod putem observa ca aceasta returneaza un numar aleator mai mic decat argumentul dat (aici a1). Am cautat mai departe unde este folosita aceasta functie, pe care am numit-o **rand_up_to**.

xrefs to rand_up_to			
Direction	Type	Address	Text
Up	p	sub_100367A+53	call rand_up_to
Up	p	sub_100367A+61	call rand_up_to

OK Cancel Search Help

Line 1 of 2

```
1 int sub_100367A()
2 {
3     signed int v0; // ebx
4     int v1; // esi
5     int v2; // eax
6     signed int v4; // [esp-4h] [ebp-10h]
7
8     dword_1005164 = 0;
9     if ( dword_10056AC != dword_1005334 || uValue != dword_1005338 )
10         v4 = 6;
11     else
12         v4 = 4;
13     v0 = v4;
14     dword_1005334 = dword_10056AC;
15     dword_1005338 = uValue;
16     sub_1002ED5();
17     dword_1005160 = 0;
18     dword_1005330 = dword_10056A4;
19     do
20     {
21         do
22         {
23             v1 = rand_up_to(dword_1005334) + 1;
24             v2 = rand_up_to(dword_1005338) + 1;
25         }
26         while ( byte_1005340[32 * v2 + v1] < 0 );
27         byte_1005340[32 * v2 + v1] |= 0x80u;
28         --dword_1005330;
29     }
30     while ( dword_1005330 );
31     dword_100579C = 0;
32     dword_1005330 = dword_10056A4;
33     dword_1005194 = dword_10056A4;
34     dword_10057A4 = 0;
35     dword_10057A0 = dword_1005334 * dword_1005338 - dword_10056A4;
36     dword_1005000 = 1;
37     sub_100346A(0);
38     return sub_1001950(v0);
39 }
```

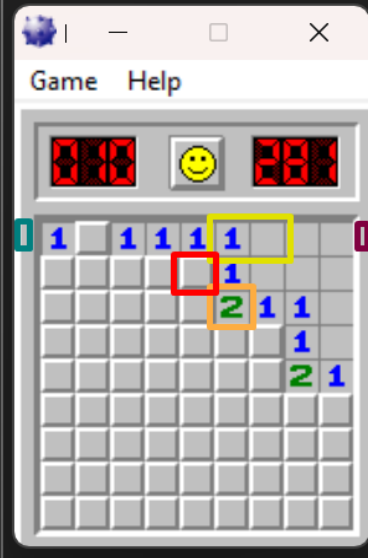
Se pare ca si aceasta este folosita intr-un singur loc. Pentru ca initial nu am stiut ce rol are aceasta functie am decis sa rulez Minesweeper in x32dbg si sa pun Breakpoints in cateva locuri din codul de mai sus, mai exact la inceputul si sfarsitul functiei si la inceputul si sfarsitul **while**-ului mare.

• 0100367A	A1 AC560001	mov eax,dword ptrds:[10056A4]
• 0100367F	8B0D A8560001	mov ecx,dword ptrds:[10056A8]
• 01003685	53	push ebx
• 01003686	56	push esi
• 01003687	57	push edi
• 01003688	33FF	xor edi,edi
• 0100368A	3B05 34530001	cmp eax,dword ptrds:[1005334]
• 01003690	893D 64510001	mov dword ptrds:[1005164],edi
• 01003696	✓ 75 0C	jne winmine.10036A4
• 01003698	3B0D 38530001	cmp ecx,dword ptrds:[1005338]
• 0100369E	✓ 75 04	jne winmine.10036A4
• 010036A0	6A 04	push 4
• 010036A2	✓ EB 02	jmp winmine.10036A6
• 010036A4	6A 06	push 6
• 010036A6	5B	pop ebx
• 010036A7	A3 34530001	mov dword ptrds:[1005334],eax
• 010036AC	890D 38530001	mov dword ptrds:[1005338],ecx
• 010036B2	E8 1EF8FFFF	call winmine.1002ED5
• 010036B7	A1 A4560001	mov eax,dword ptrds:[10056A4]
• 010036BC	893D 60510001	mov dword ptrds:[1005164],edi
• 010036C2	A3 30530001	mov dword ptrds:[1005330],eax
• 010036C7	FF35 34530001	push dword ptrds:[1005334]
• 010036CD	E8 6E020000	call <winmine.rand_up_to>
• 010036D2	FF35 38530001	push dword ptrds:[1005338]
• 010036D8	8BF0	mov esi,eax
• 010036DA	46	inc esi
• 010036DB	E8 60020000	call <winmine.rand_up_to>
• 010036E0	40	inc eax
• 010036E1	8BC8	mov ecx,eax
• 010036E3	C1E1 05	shl ecx,5
• 010036E6	F68431 40530001 80	test byte ptrds:[ecx+esi+1005340],80
• 010036EE	^ 75 D7	jne winmine.10036C7
• 010036F0	C1E0 05	shl eax,5
• 010036F3	8D8430 40530001	lea eax,dword ptrds:[eax+esi+1005340]
• 010036FA	8008 80	or byte ptrds:[eax],80
• 010036FD	FF0D 30530001	dec dword ptrds:[1005330]
• 01003703	^ 75 C2	jne winmine.10036C7
• 01003705	8B0D 38530001	mov ecx,dword ptrds:[1005338]
• 0100370B	0FAF0D 34530001	imul ecx,dword ptrds:[1005334]
• 01003712	A1 A4560001	mov eax,dword ptrds:[10056A4]
• 01003717	2BC8	sub ecx,eax
• 01003719	57	push edi
• 0100371A	893D 9C570001	mov dword ptrds:[1005790],edi
• 01003720	A3 30530001	mov dword ptrds:[1005330],eax
• 01003725	A3 94510001	mov dword ptrds:[1005194],eax
• 0100372A	893D A4570001	mov dword ptrds:[10057A4],edi
• 01003730	890D A0570001	mov dword ptrds:[10057A0],ecx
• 01003736	C705 00500001 010000	mov dword ptrds:[1005000],1
• 01003740	E8 25FDFFFF	call winmine.100346A
• 01003745	53	push ebx
• 01003746	E8 05E2FFFF	call winmine.1001950
• 0100374B	5F	pop edi
• 0100374C	5E	pop esi
• 0100374D	5B	pop ebx
• 0100374E	C3	ret
• 0100374F	53	push ebx

Cand programul a ajuns la inceputul while-ului am mutat EIP dupa acesta. Aceasta actiune a rezultat intr-un camp gol in joc. De asemenea, resetarea nivelului prin apasarea fetei zambitoare rezulta in redeclansarea acestor breakpointuri. De aici am ajuns la concluzia ca acea parte este esentiala la initializarea casutelor.

In urmatoarele poze am corelat datele din memorie cu casutele din joc.

Address	Unsigned byte (8-bit)							
01005360	16	65	143	65	65	65	65	64
01005368	64	64	16	15	15	15	15	15
01005370	15	15	15	15	15	15	15	15
01005378	15	15	15	15	15	15	15	15
01005380	16	15	15	15	15	143	65	64
01005388	64	64	16	15	15	15	15	15
01005390	15	15	15	15	15	15	15	15
01005398	15	15	15	15	15	15	15	15
010053A0	16	15	15	15	15	15	66	65
010053A8	65	64	16	15	15	15	15	15
010053B0	15	15	15	15	15	15	15	15
010053B8	15	15	15	15	15	15	15	15
010053C0	16	143	15	15	15	15	15	143
010053C8	65	64	16	15	15	15	15	15
010053D0	15	15	15	15	15	15	15	15
010053D8	15	15	15	15	15	15	15	15
010053E0	16	143	15	15	15	15	15	15
010053E8	66	65	16	15	15	15	15	15
010053F0	15	15	15	15	15	15	15	15
010053F8	15	15	15	15	15	15	15	15
01005400	16	15	15	15	15	15	15	15
01005408	15	143	16	15	15	15	15	15
01005410	15	15	15	15	15	15	15	15
01005418	15	15	15	15	15	15	15	15
01005420	16	143	15	143	15	15	15	15
01005428	15	143	16	15	15	15	15	15



Address	Unsigned byte (8-bit)							
01005360	16	65	143	65	65	65	65	64
01005368	64	64	16	15	15	15	15	15
01005370	15	15	15	15	15	15	15	15
01005378	15	15	15	15	15	15	15	15
01005380	16	14	13	15	15	143	65	64
01005388	64	64	16	15	15	15	15	15
01005390	15	15	15	15	15	15	15	15
01005398	15	15	15	15	15	15	15	15
010053A0	16	15	15	15	15	15	66	65
010053A8	65	64	16	15	15	15	15	15
010053B0	15	15	15	15	15	15	15	15
010053B8	15	15	15	15	15	15	15	15
010053C0	16	143	15	15	15	15	15	143
010053C8	65	64	16	15	15	15	15	15
010053D0	15	15	15	15	15	15	15	15
010053D8	15	15	15	15	15	15	15	15
010053E0	16	143	15	15	15	15	15	15
010053E8	66	65	16	15	15	15	15	15
010053F0	15	15	15	15	15	15	15	15
010053F8	15	15	15	15	15	15	15	15
01005400	16	15	15	15	15	15	15	15
01005408	15	143	16	15	15	15	15	15
01005410	15	15	15	15	15	15	15	15
01005418	15	15	15	15	15	15	15	15
01005420	16	143	15	143	15	15	15	15
01005428	15	143	16	15	15	15	15	15



Address	Unsigned byte (8-bit)							
01005360	16	65	142	65	65	65	65	64
01005368	64	64	16	15	15	15	15	15
01005370	15	15	15	15	15	15	15	15
01005378	15	15	15	15	15	15	15	15
01005380	16	15	15	15	15	15	65	64
01005388	64	64	16	15	15	15	15	15
01005390	15	15	15	15	15	15	15	15
01005398	15	15	15	15	15	15	15	15
010053A0	16	15	15	15	15	15	66	65
010053A8	65	64	16	15	15	15	15	15
010053B0	15	15	15	15	15	15	15	15
010053B8	15	15	15	15	15	15	15	15
010053C0	16	143	15	15	15	15	15	143
010053C8	65	64	16	15	15	15	15	15
010053D0	15	15	15	15	15	15	15	15
010053D8	15	15	15	15	15	15	15	15
010053E0	16	143	15	15	15	15	15	15
010053E8	66	65	16	15	15	15	15	15
010053F0	15	15	15	15	15	15	15	15
010053F8	15	15	15	15	15	15	15	15
01005400	16	15	15	15	15	15	15	15
01005408	15	143	16	15	15	15	15	15
01005410	15	15	15	15	15	15	15	15
01005418	15	15	15	15	15	15	15	15
01005420	16	143	15	143	15	15	15	15
01005428	15	143	16	15	15	15	15	15



De aici am ajuns la urmatoarele concluzii:

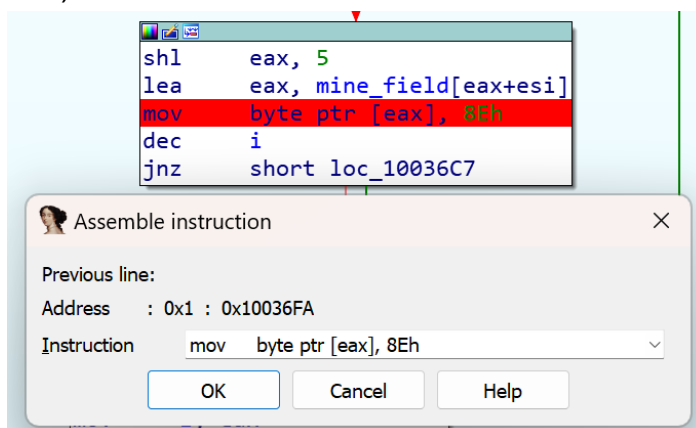
- 15 = casuta neapasata
- 14 = casuta neapasata cu stegulet
- 13 = casuta neapasata cu semnul intrebarii
- 64 = casuta apasata cu 0 bombe in vecinatate
- 65 = casuta apasata cu o bomba in vecinatate
- etc... (64 + numarul de bombe din vecinatate)
- 143 = bomba
- 142 = bomba cu stegulet
- 141 = bomba cu semnul intrebarii

Ba chiar mai mult, aceste date au mai mult sens daca le gandim ca **flags** in binar sau hexadecimal. Fiecare casuta e reprezentata de cate un byte astfel:

Handwritten notes on a black background showing the binary representation of flags for different mine states:

- Baza = 0000 1111 = 0x0F
- Bombă = 1 - - - - - => 0x80
- Steag = - - - - - 10 => 0x0E
- Semnul/? = - - - - - 01 => 0x0B
- Apăsat = - 1 - - - - => 0x40

In functia de generare a campului se executa operatia **OR** cu **0x80** pe matrice. Acum ne putem da seama ca asta inseamna de fapt adaugarea unei bombe pe acea pozitie. Pentru a avea stegulete pe toate bombele, am schimbat in IDA instructiunea de **OR** cu setarea in memorie a valorii **0x8E** (casuta de baza + bomba + steag = **1000 1110**).

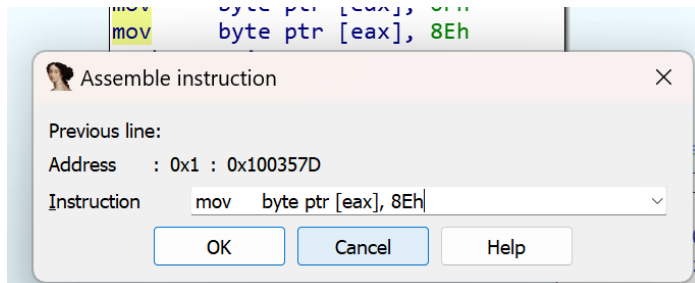


Succes!



Exista insa si cazul in care apasam din prima pe o bomba, iar programul o repositioneaza. Am gasit codul care seteaza noua bomba, insa modificarea acestuia la fel ca in cazul precedent nu a dus la acelasi rezultat. Casuta avea setata valoarea corecta (0x8E) si nu se putea apasa (casutele cu stegulet nu pot fi apasate), insa stegulețul nu era si afisat.

Instructiunea este la adresa **0x100357D**.



2. Semnul intrebarii pe casutele goale

Cautand prin locurile in care se foloseste matricea am gasit portiunea de initializare acesteia cu valoarea 15 (**0x0F**). Am inlocuit aceasta valoare cu 13 (**0x0D**), iar rezultatul a fost ca toate celelalte patratele aveau semnul intrebarii pe ele.

Urmatorul pas era sa adaug 1 (sau sa setez valoarea **0x0E**) pe casutele adiacente bombelor. Acest lucru l-as fi facut la setarea bombelor, insa nu am reusit sa fac acest lucru.



3. Highscores

In mod implicit, scorurile sunt de 999 secunde de catre “Anonymous”. Am incercat mai multe metode de a gasi ori cand se completeaza aceste valori, ori unde sunt stocate valorile initiale.

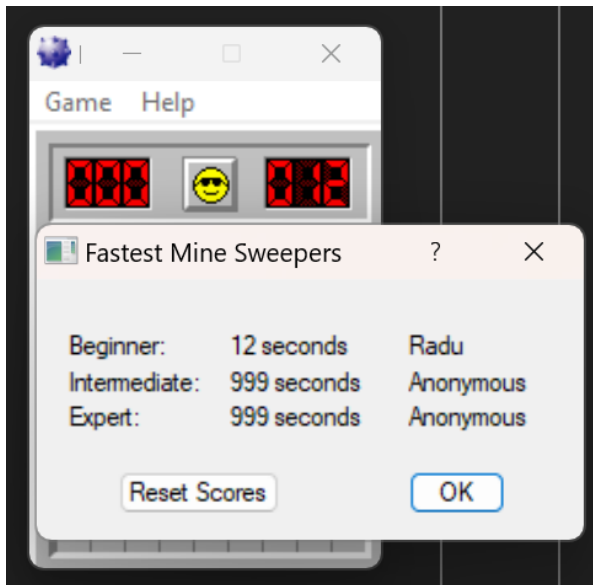
Am cautat dupa stringul “Anonymous”:

All Modules (Strings)			
Address	Disassembly	String /	String
759DADEB	push rcx, 759A905C	759A905C	L"anonymous"
75D02DEE	mov dword ptr ss:[ebp-40], ucrtbase.75CA8DC0	75CA8DC0	"anonymous namespace"
774EF7B5	push msvcrt.774B452C	774B452C	"anonymous namespace"
774F0071	push msvcrt.774B452C	774B452C	"anonymous namespace"

Am cautat cand se creaza o noua fereastră, incercand sa gasesc pop up-ul in care utilizatorul isi trece numele:

Address	Ordinal	Name	Library
010010B0		GetDesktopWindow	USER32
010010E0		MapWindowPoints	USER32
01001118		MoveWindow	USER32
01001124		DefWindowProcW	USER32
01001134		ShowWindow	USER32
01001158		UpdateWindow	USER32
0100115C		CreateWindowExW	USER32

Insa nu am gasit informatii utile. Am terminat jocul si mi-am completat numele. Apoi am cautat referinte catre numele meu in x32dbg.



All Modules (Strings)			
Address	Disassembly	String /	String
0100176B	push winmine.10056D8	010056D8	L"Radu"
01001799	push winmine.10056D8	010056D8	L"Radu"
01001856	mov eax,winmine.10056D8	010056D8	L"Radu"
010018EE	mov eax,winmine.10056D8	010056D8	L"Radu"
01002CC0	push winmine.10056D8	010056D8	L"Radu"
01002E78	push winmine.10056D8	010056D8	L"Radu"
01003C3C	push winmine.10056D8	010056D8	L"Radu"

Am gasit un loc interesant in care se foloseste acest string si am analizat functia in IDA.

0100176A	57	push edi	
0100176B	68 D8560001	push <winmine.highscore_name>	10056D8:L"Radu"
01001770	A3 D4560001	mov dword ptr ds:[10056D4],eax	
01001775	A3 D0560001	mov dword ptr ds:[10056D0],eax	
0100177A	A3 CC560001	mov dword ptr ds:[10056CD],eax	
0100177F	FFD6	call esi	
01001781	57	push edi	
01001782	68 18570001	push winmine.1005718	1005718:L"Anonymous"
01001787	FFD6	call esi	
01001789	57	push edi	
0100178A	53	push ebx	
0100178B	FFD6	call esi	
0100178D	5F	pop edi	
0100178E	C705 5C510001 010001	mov dword ptr ds:[100515C],1	
01001798	5E	pop esi	
01001799	68 D8560001	push <winmine.highscore_name>	10056D8:L"Radu"
0100179F	5735 C6560001	push dword ptr ds:[10056CD]	

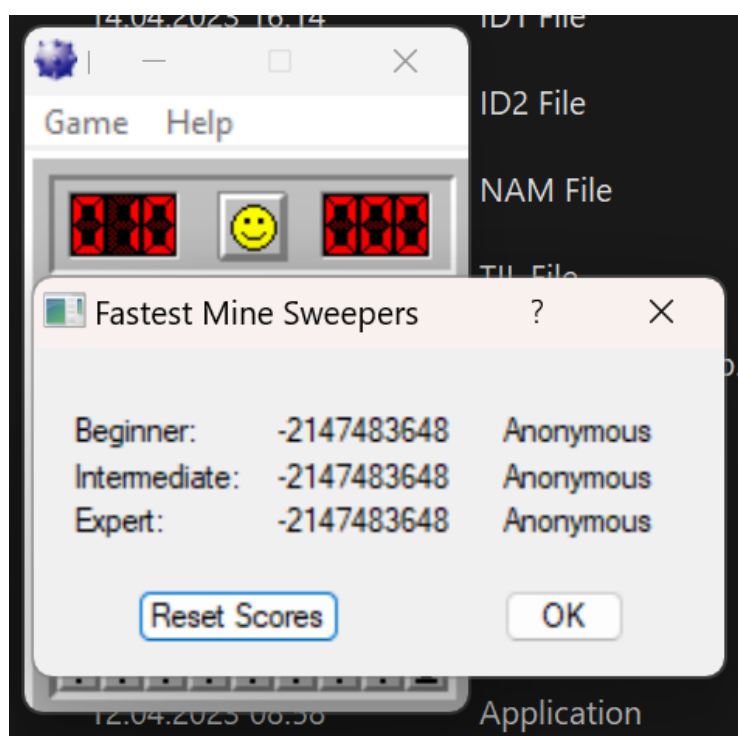
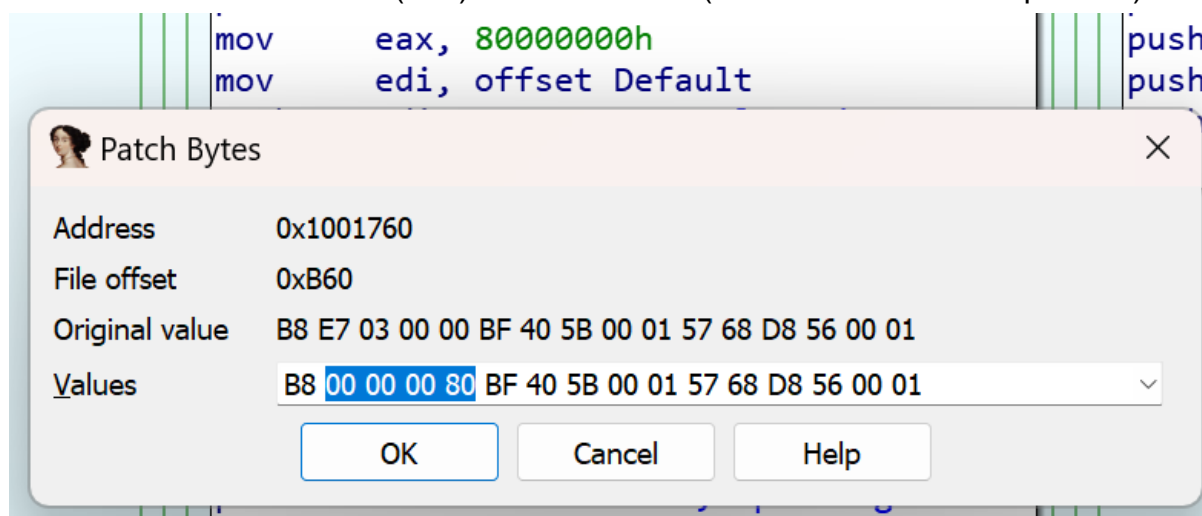
```

^(_DWORD ^)&word_10056D0 = 999;
*(_DWORD *)&word_10056CC = 999;
lstrcpyW(&ReturnedString, &Default);
lstrcpyW(&word_1005718, &Default);
lstrcpyW(&word_1005758, &Default);
dword_100515C = 1;
goto LABEL_11;
}

```

Aici se initializeaza datele din **Highscores** (se poate observa si valoarea 999 de mai sus).

Pentru a modifica valoarea scorurilor initiale am facut un patch in IDA. Am setat valoarea de la **0x3E7** (999) la **0x80000000** (cea mai mica valoare posibila).

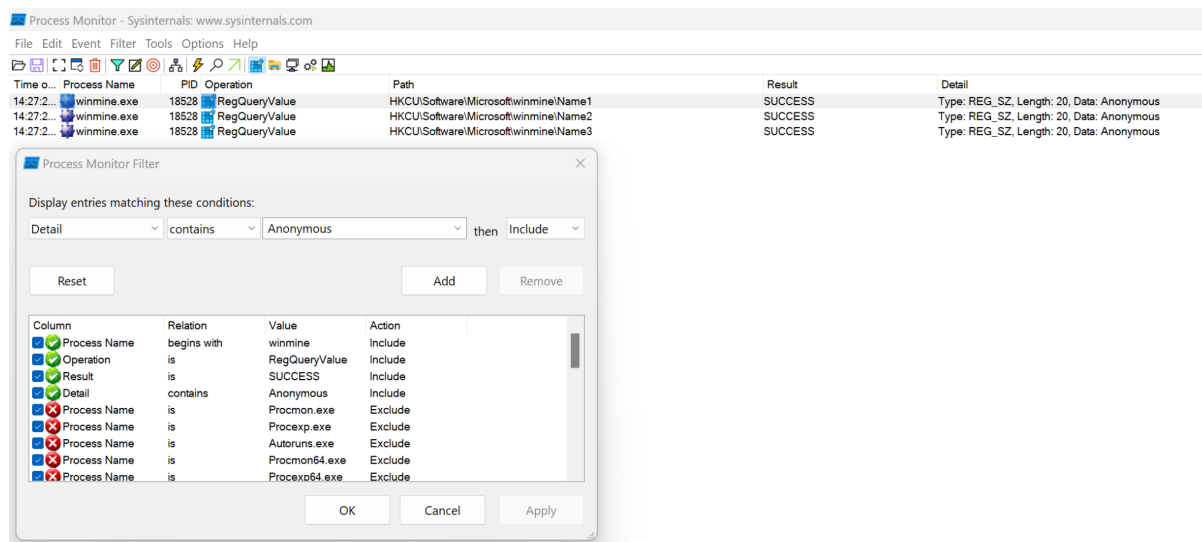


Din pacate, la repornirea jocului scorurile sunt aduse la valoarea 0 daca sunt negative. Probabil putem modifica acest comportament, dar nu am incercat acest lucru.

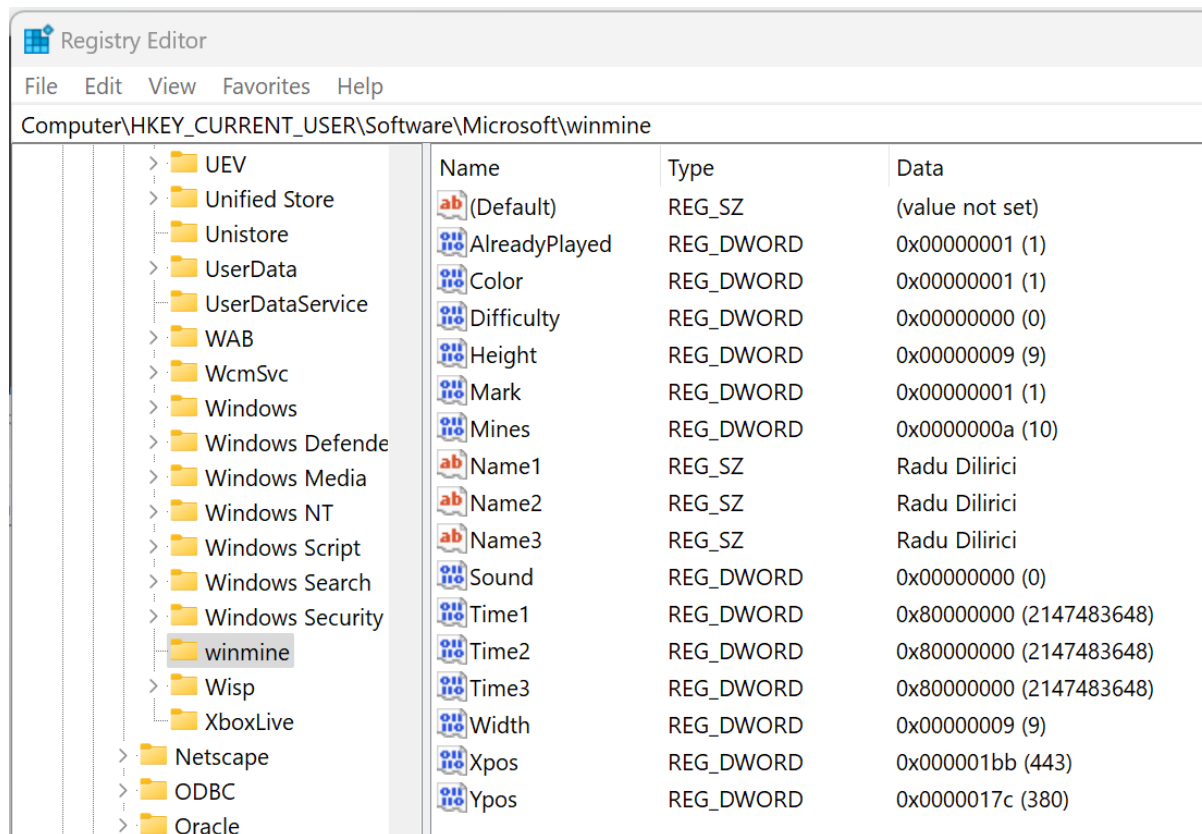
Cat despre nume, nu am gasit in program valoarea default a acestuia. Am gasit intr-un blog cum aceste date ar fi stocate in registrii:

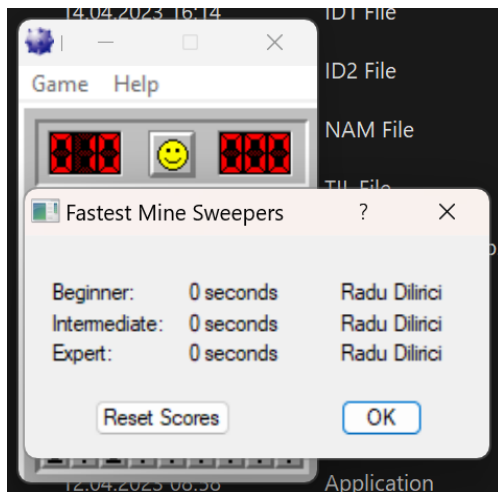
<https://www.unknowncheats.me/forum/general-programming-and-reversing/18330-reversing-minesweeper-hidden-options.html>. Asa ca am urmat aceasta idee.

Intr-adevar, dupa analiza programului cu **Process Monitor**, am putut observa ce registrii sunt folositi pentru a stoca datele respective.



Apoi am intrat in **Registry Editor** si am modificat valorile (Name1, Name2 si Name3) din acel loc.



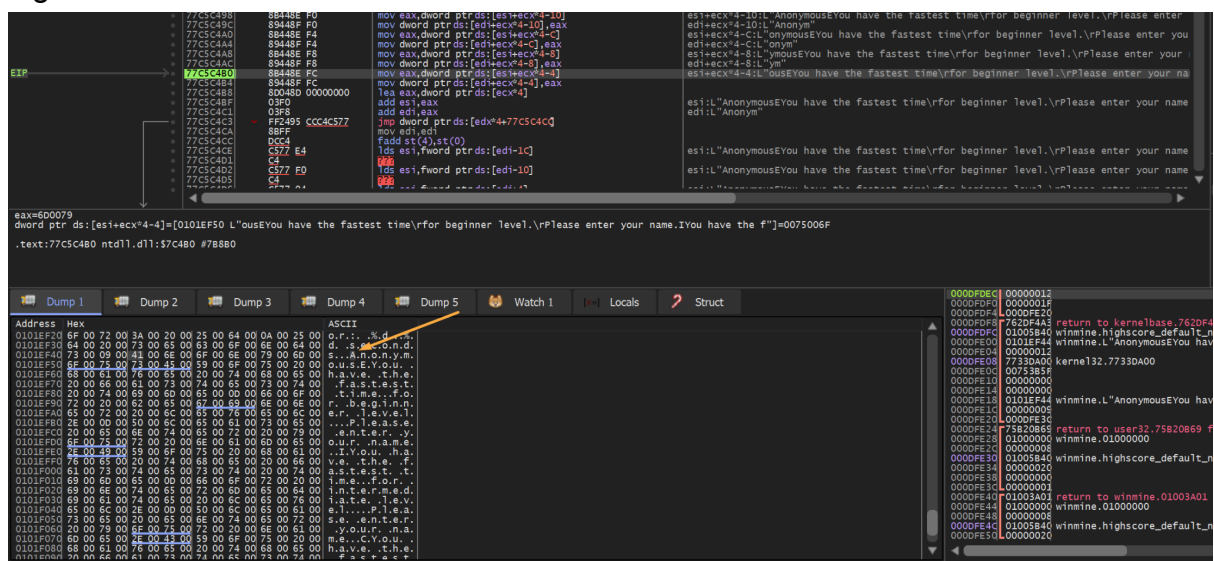


Acest lucru insa modifica datele doar temporar. La resetarea scorurilor, acestea vor avea din nou numele “Anonymous”.

Cred ca am gasit functia in care initializeaza aceste valori, insa nu am stiut cum sa o modific astfel incat sa contina stringul dorit.

```
1 int __stdcall initialize_default_highscore_name(__int16 a1, LPWSTR lpBuffer, int cchBufferMax)
2 {
3     int result; // eax
4
5     result = LoadStringW(hModule, (unsigned __int16)a1, lpBuffer, cchBufferMax);
6     if (!result)
7         result = sub_1003950(1001);
8     return result;
9 }
```

Pare ca numele “Anonymous” sunt luate din memoria programului, dar din nou, nu am reusit sa gasesc locul in care aceasta memorie este initializata. Problema principala a fost alocarea dinamica a acestei memorii. Informatia nu se regaseste mereu la aceeasi adresa.



Pe langa cuvantul “*Anonymous*” subliniat mai jos (de la acea adresa este luat) se mai observa si alte texte ale programului, precum “*Unable to allocate a timer*. *Please exit some of your applications and try again.*”. Nu am reusit sa gasesc referinte la aceste texte, nici in ce locuri apar in joc.

Dump 1				Dump 2				Dump 3				Dump 4					
Address	Hex								ASCII								
0101EDF8	72	00	61	00	6E	00	73	00	6C	00	61	00	74	00	69	00	r.a.n.s.l.a.t.i.
0101EE08	6F	00	6E	00	00	00	00	00	09	04	B0	04	00	00	00	00	o.n.....
0101EE18	00	00	0B	00	4D	00	69	00	6E	00	65	00	73	00	77	00	...M.i.n.e.s.w.
0101EE28	65	00	65	00	70	00	65	00	72	00	00	00	11	00	4D	00	e.e.p.e.r....M.
0101EE38	69	00	6E	00	65	00	73	00	77	00	65	00	65	00	70	00	i.n.e.s.w.e.e.p.
0101EE48	65	00	72	00	20	00	45	00	72	00	72	00	6F	00	72	00	e.r. .E.r.r.o.r.
0101EE58	51	00	55	00	6E	00	61	00	62	00	6C	00	65	00	20	00	Q.U.n.a.b.l.e. .
0101EE68	74	00	6F	00	20	00	61	00	6C	00	6C	00	6F	00	63	00	t.o. .a.l.l.o.c.
0101EE78	61	00	74	00	65	00	20	00	61	00	20	00	74	00	69	00	a.t.e. .a. .t.i.
0101EE88	6D	00	65	00	72	00	2E	00	20	00	20	00	50	00	6C	00	m.e.r... .P.l.
0101EE98	65	00	61	00	73	00	65	00	20	00	65	00	78	00	69	00	e.a.s.e. .e.x.i.
0101EEA8	74	00	20	00	73	00	6F	00	6D	00	65	00	20	00	6F	00	t. .s.o.m.e. .o.
0101EEB8	66	00	20	00	79	00	6F	00	75	00	72	00	20	00	61	00	f. .y.o.u.r. .a.
0101EEC8	70	00	70	00	6C	00	69	00	63	00	61	00	74	00	69	00	p.p.l.i.c.a.t.i.
0101EED8	6F	00	6E	00	73	00	20	00	61	00	6E	00	64	00	20	00	o.n.s. .a.n.d. .
0101EEE8	74	00	72	00	79	00	20	00	61	00	67	00	61	00	69	00	t.r.y. .a.g.a.i.
0101EEF8	6E	00	2E	00	0D	00	4F	00	75	00	74	00	20	00	6F	00	n....O.u.t. .o.
0101EF08	66	00	20	00	4D	00	65	00	6D	00	6F	00	72	00	79	00	f. .M.e.m.o.r.y.
0101EF18	09	00	45	00	72	00	72	00	6F	00	72	00	3A	00	20	00	.E.r.r.o.r.: .
0101EF28	25	00	64	00	0A	00	25	00	64	00	20	00	73	00	65	00	%d...%d...s.c.
0101EF38	63	00	6F	00	6E	00	64	00	73	00	09	00	41	00	6E	00	c.o.n.d.s...A.n.
0101EF48	6F	00	6E	00	79	00	6D	00	6F	00	75	00	73	00	45	00	o.n.y.m.o.u.s.E.
0101EF58	59	00	6F	00	75	00	20	00	68	00	61	00	76	00	65	00	t.o.u. .h.a.v.e.
0101EF68	20	00	74	00	68	00	65	00	20	00	66	00	61	00	73	00	.t.h.e. .f.a.s.
0101EF78	74	00	65	00	73	00	74	00	20	00	74	00	69	00	6D	00	t.e.s.t. .t.i.m.
0101EF88	65	00	0D	00	66	00	6F	00	72	00	20	00	62	00	65	00	e...f.o.r. .b.e.
0101EF98	67	00	69	00	6E	00	6E	00	65	00	72	00	20	00	6C	00	g.i.n.n.e.r. .l.
0101EFA8	65	00	76	00	65	00	6C	00	2E	00	0D	00	50	00	6C	00	e.v.e.l....P.l.
0101EFB8	65	00	61	00	73	00	65	00	20	00	65	00	6E	00	74	00	e.a.s.e. .e.n.t.
0101EFC8	65	00	72	00	20	00	79	00	6F	00	75	00	72	00	20	00	e.r. .y.o.u.r. .
0101EFD8	6E	00	61	00	6D	00	65	00	2E	00	49	00	59	00	6F	00	n.a.m.e...I.Y.o.
0101EFE8	75	00	20	00	68	00	61	00	76	00	65	00	20	00	74	00	u. .h.a.v.e. .t.
0101EFF8	68	00	65	00	20	00	66	00	61	00	73	00	74	00	65	00	h.e. .f.a.s.t.e.
0101F008	73	00	74	00	20	00	74	00	69	00	6D	00	65	00	0D	00	s.t. .t.i.m.e...
0101F018	66	00	6F	00	72	00	20	00	69	00	6E	00	74	00	65	00	f.o.r. .i.n.t.e.
0101F028	72	00	6D	00	65	00	64	00	69	00	61	00	74	00	65	00	r.m.e.d.i.a.t.e.
0101F038	20	00	6C	00	65	00	76	00	65	00	6C	00	2E	00	0D	00	.l.e.v.e.l....
0101F048	50	00	6C	00	65	00	61	00	73	00	65	00	20	00	65	00	P.l.e.a.s.e. .e.
0101F058	6E	00	74	00	65	00	72	00	20	00	79	00	6F	00	75	00	n.t.e.r. .v.o.u.

4. Creator

Am gasit numele celor doi autori in acelasi loc din memoria programuli. Am schimbat memoria astfel incat sa reprezinte numele meu, iar astfel a ajuns sa fie afisat in pagina de **About**.

0101EF80	20 00 74 00	69 00 6D 00	65 00 0D 00	66 00 6F 00	.t.i.m.e...f.o.
0101EF90	72 00 20 00	62 00 65 00	67 00 69 00	6E 00 6E 00	r. .b.e.g.i.n.n.
0101EFA0	65 00 72 00	20 00 6C 00	65 00 76 00	65 00 6C 00	e.r. .l.e.v.e.l.
0101EFB0	2E 00 0D 00	50 00 6C 00	65 00 61 00	73 00 65 00	...P.l.e.a.s.e.
0101EFC0	20 00 65 00	6E 00 74 00	65 00 72 00	20 00 79 00	.e.n.t.e.r. .y.
0101EFD0	6F 00 75 00	72 00 20 00	6E 00 61 00	6D 00 65 00	o.u.r. .n.a.m.e.
0101EFE0	2E 00 49 00	59 00 6F 00	75 00 20 00	68 00 61 00	..I.Y.o.u. .h.a.
0101EFF0	76 00 65 00	20 00 74 00	68 00 65 00	20 00 66 00	v.e. .t.h.e. .f.
0101F000	61 00 73 00	74 00 65 00	73 00 74 00	20 00 74 00	a.s.t.e.s.t. .t.
0101F010	69 00 6D 00	65 00 0D 00	66 00 6F 00	72 00 20 00	i.m.e...f.o.r. .
0101F020	69 00 6E 00	74 00 65 00	72 00 6D 00	65 00 64 00	i.n.t.e.r.m.e.d.
0101F030	69 00 61 00	74 00 65 00	20 00 6C 00	65 00 76 00	i.a.t.e. .l.e.v.
0101F040	65 00 6C 00	2E 00 0D 00	50 00 6C 00	65 00 61 00	e.l....P.l.e.a.
0101F050	73 00 65 00	20 00 65 00	6E 00 74 00	65 00 72 00	s.e. .e.n.t.e.r.
0101F060	20 00 79 00	6F 00 75 00	72 00 20 00	6E 00 61 00	.y.o.u.r. .n.a.
0101F070	6D 00 65 00	2E 00 43 00	59 00 6F 00	75 00 20 00	m.e...C.Y.o.u. .
0101F080	68 00 61 00	76 00 65 00	20 00 74 00	68 00 65 00	h.a.v.e. .t.h.e.
0101F090	20 00 66 00	61 00 73 00	74 00 65 00	73 00 74 00	.f.a.s.t.e.s.t.
0101F0A0	20 00 74 00	69 00 6D 00	65 00 0D 00	66 00 6F 00	.t.i.m.e...f.o.
0101F0B0	72 00 20 00	65 00 78 00	70 00 65 00	72 00 74 00	r. .e.x.p.e.r.t.
0101F0C0	20 00 6C 00	65 00 76 00	65 00 6C 00	2E 00 0D 00	.l.e.v.e.l....
0101F0D0	50 00 6C 00	65 00 61 00	73 00 65 00	20 00 65 00	P.l.e.a.s.e. .e.
0101F0E0	6E 00 74 00	65 00 72 00	20 00 79 00	6F 00 75 00	n.t.e.r. .y.o.u.
0101F0F0	72 00 20 00	6E 00 61 00	6D 00 65 00	2E 00 0B 00	r. .n.a.m.e....
0101F100	4D 00 69 00	6E 00 65 00	73 00 77 00	65 00 65 00	M.i.n.e.s.w.e.e.
0101F110	70 00 65 00	72 00 21 00	62 00 79 00	20 00 52 00	p.e.r!.b.y. .R.
0101F120	6F 00 62 00	65 00 72 00	74 00 20 00	44 00 6F 00	o.b.e.r.t. .D.o.
0101F130	6E 00 6E 00	65 00 72 00	20 00 61 00	6E 00 64 00	n.n.e.r. .a.n.d.
0101F140	20 00 43 00	75 00 72 00	74 00 20 00	4A 00 6F 00	.C.u.r.t. .J.o.
0101F150	68 00 6E 00	73 00 6F 00	6E 00 00 00	00 00 00 00	h.n.s.o.n.....
0101F160	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F170	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F180	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F190	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F1A0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F1B0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

0101F0F0	72 00 20 00	6E 00 61 00	6D 00 65 00	2E 00 0B 00	r. .n.a.m.e....
0101F100	4D 00 69 00	6E 00 65 00	73 00 77 00	65 00 65 00	M.i.n.e.s.w.e.e.
0101F110	70 00 65 00	72 00 21 00	62 00 79 00	20 00 52 00	p.e.r!.b.y. .R.
0101F120	61 00 64 00	75 00 20 00	44 00 69 00	6C 00 69 00	a.d.u. .D.i.l.i.
0101F130	72 00 69 00	63 00 69 00	00 00 00 00	00 00 00 00	r.i.c.i.....
0101F140	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F150	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F160	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0101F170	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00



A se ignora iconita neagra din poza, pe care am stricat-o pe parcursul modificarii programului. In executabilul trimis, iconita este cea originala.

Acesta metoda m-a ajutat in a-mi scrie numele in aceasta pagina, inasa este o modificare temporara. La repornirea jocului, numele originale sunt din nou folosite.

Pasii prezentati sunt doar care au adus rezultate. Am incercat mai multe abordari fara a progresa in rezolvarea cerintelor.