Lab 7. Hacking Minesweeper with Ollydbg

What You Need

A Windows 2016 machine, real or virtual. Other Windows versions should also work.

Purpose

To hack MineSweeper at the binary level. This gives you practice using the Ollydbg debugger, Procdump, and Python.

Downloading OllyDbg

If you don't already have it, download OllyDbg 1.10 here:

http://www.ollydbg.de/

Right-click the file and click **Extract**, "**Extract All...**".

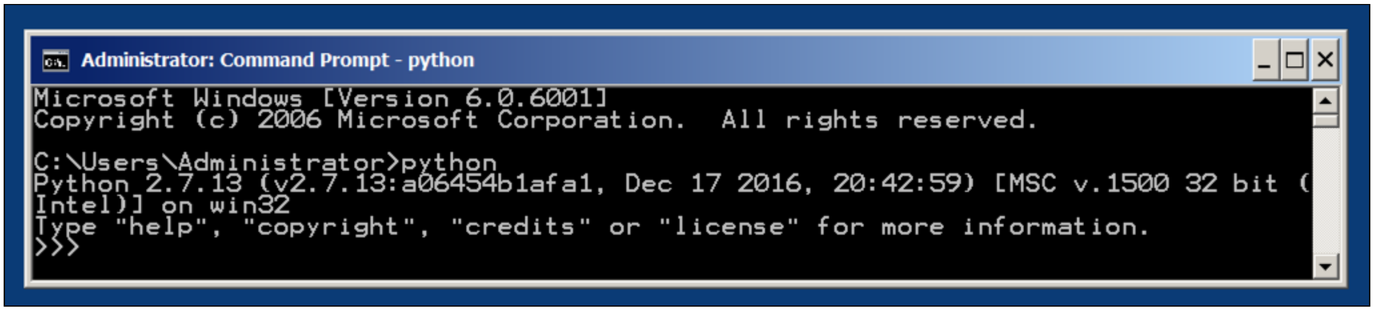
Double-click the red icon to launch it.

Testing Python

To see if python 2.7 is already installed, open a Command Prompt and execute this command:

```
python
```

You should see a "Python 2.7" message, as shown below.



```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>python
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If python does not open, follow these instructions to install it:

https://samsclass.info/124/proj14/python2.7-win.htm

Getting Minesweeper

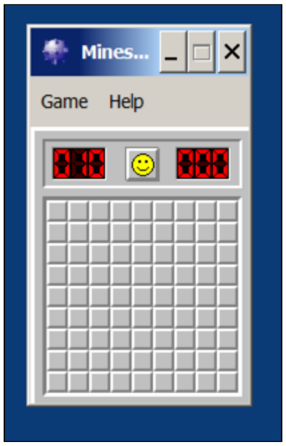
Download the minesweeper program from the link below.

minesam.exe.zip

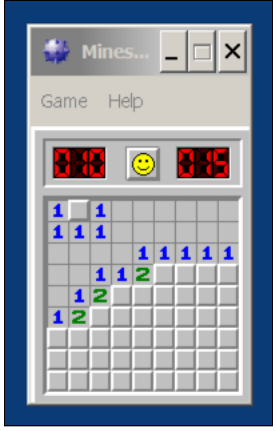
Right-click the zipped file and click "**Extract All...**", **Extract**.

Double-click the **minesam.exe** file to launch Minesweeper.

The game launches. Click **Game**, **Beginner** to see the small gameboard shown below. as shown below.



Click a cell. Some of the cells appear empty, and others are revealed with numbers in them, as shown below.

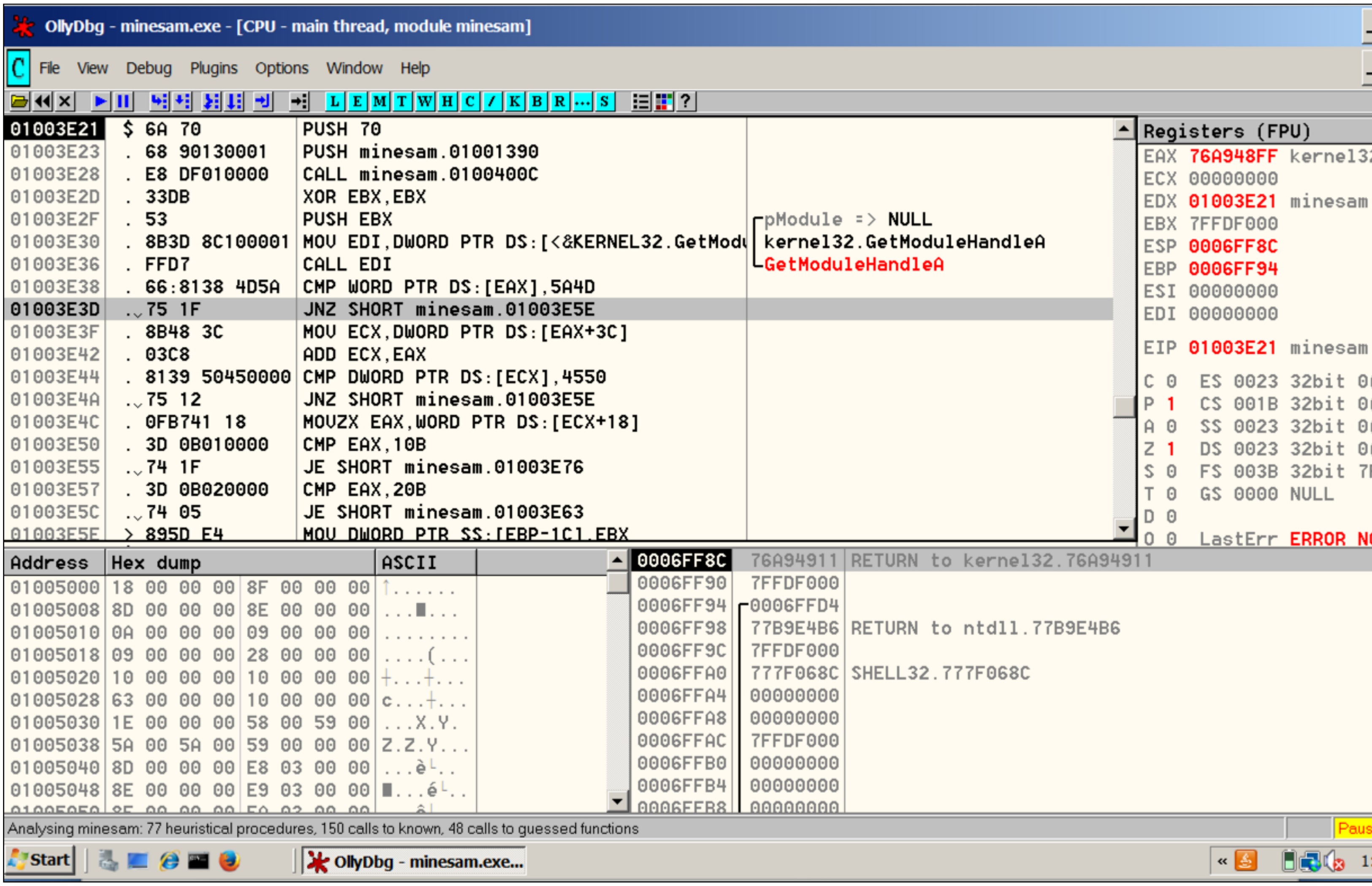


Viewing the Game in OllyDbg

Close Minesweeper.

Launch OllyDbg. Click **File**, **Open** and open **minesam.exe**.

The program loads and pauses, as shown below.


OllyDbg - minesam.exe - [CPU - main thread, module minesam]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Start
OllyDbg - minesam.exe...
Paus

```
01003E21  $ 6A 70          PUSH 70
01003E23  . 68 90130001    PUSH minesam.01001390
01003E28  . E8 DF010000    CALL minesam.0100400C
01003E2D  . 33DB           XOR EBX,EBX
01003E2F  . 53             PUSH EBX                                 ┌pModule => NULL
01003E30  . 8B3D 8C100001  MOV EDI,DWORD PTR DS:[<&KERNEL32.GetModu │kernel32.GetModuleHandleA
01003E36  . FFD7           CALL EDI                                 └GetModuleHandleA
01003E38  . 66:8138 4D5A   CMP WORD PTR DS:[EAX],5A4D
01003E3D  .,75 1F          JNZ SHORT minesam.01003E5E
01003E3F  . 8B48 3C        MOV ECX,DWORD PTR DS:[EAX+3C]
01003E42  . 03C8           ADD ECX,EAX
01003E44  . 8139 50450000  CMP DWORD PTR DS:[ECX],4550
01003E4A  .,75 12          JNZ SHORT minesam.01003E5E
01003E4C  . 0FB741 18      MOVZX EAX,WORD PTR DS:[ECX+18]
01003E50  . 3D 0B010000    CMP EAX,10B
01003E55  .,74 1F          JE SHORT minesam.01003E76
01003E57  . 3D 0B020000    CMP EAX,20B
01003E5C  .,74 05          JE SHORT minesam.01003E63
01003E5E  > 895D E4        MOV DWORD PTR SS:[EBP-1C],EBX
```

```
Registers (FPU)
EAX 76A948FF kernel3
ECX 00000000
EDX 01003E21 minesam
EBX 7FFDF000
ESP 0006FF8C
EBP 0006FF94
ESI 00000000
EDI 00000000

EIP 01003E21 minesam

C 0  ES 0023 32bit 0
P 1  CS 001B 32bit 0
A 0  SS 0023 32bit 0
Z 1  DS 0023 32bit 0
S 0  FS 003B 32bit 7
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR N
```

```
Address   Hex dump                 ASCII
01005000  18 00 00 00 8F 00 00 00  ↑.......
01005008  8D 00 00 00 8E 00 00 00  ...█...
01005010  0A 00 00 00 09 00 00 00  ........
01005018  09 00 00 00 28 00 00 00  ....(...
01005020  10 00 00 00 10 00 00 00  ┼...┼...
01005028  63 00 00 00 10 00 00 00  c...┼...
01005030  1E 00 00 00 58 00 59 00  ...X.Y.
01005038  5A 00 5A 00 59 00 00 00  Z.Z.Y...
01005040  8D 00 00 00 E8 03 00 00  ...è└..
01005048  8E 00 00 00 E9 03 00 00  █...é└..
```

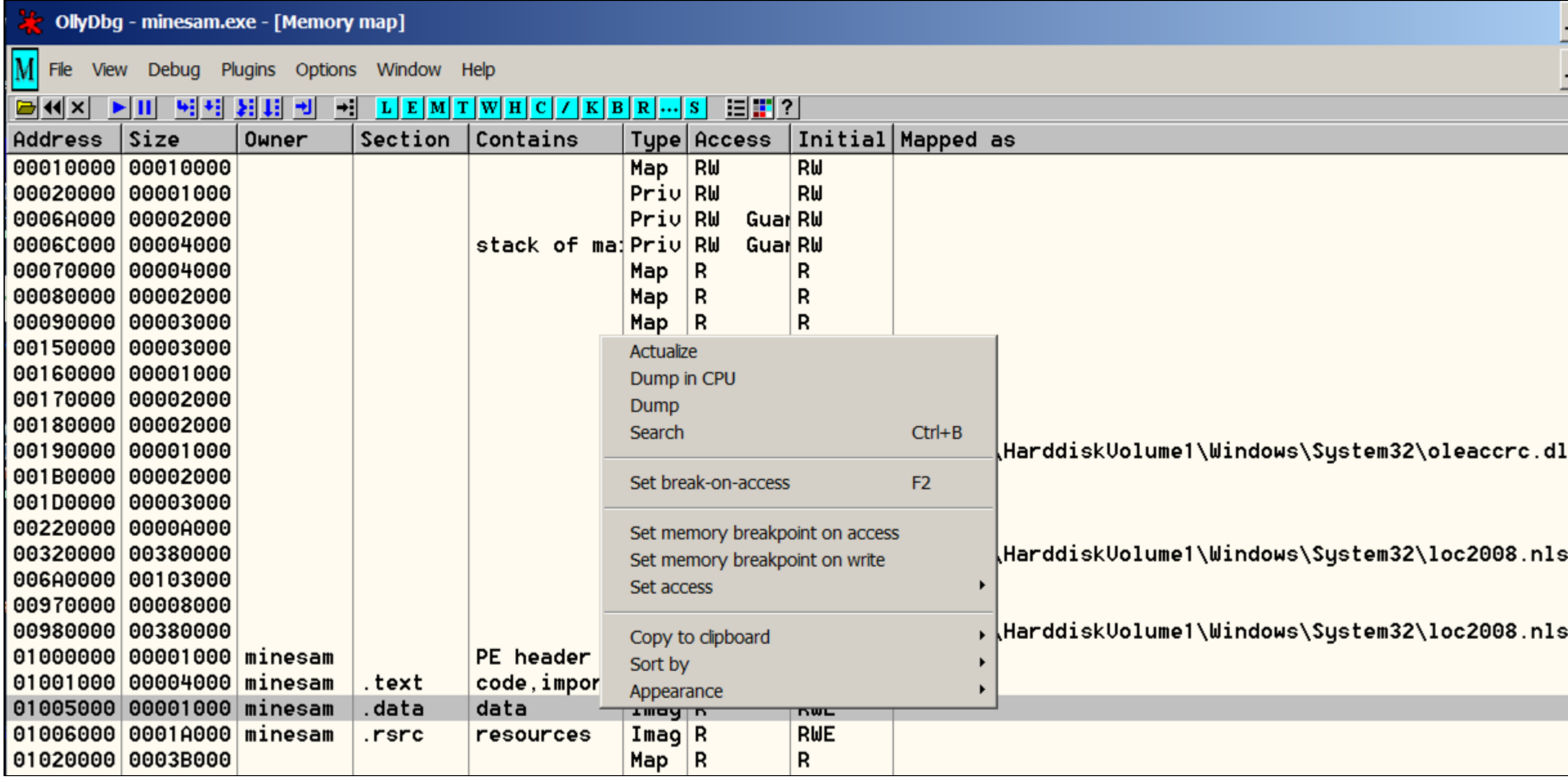
```
0006FF8C   76A94911  RETURN to kernel32.76A94911
0006FF90   7FFDF000
0006FF94  ┌0006FFD4
0006FF98  │77B9E4B6  RETURN to ntdll.77B9E4B6
0006FF9C  │7FFDF000
0006FFA0  │777F068C  SHELL32.777F068C
0006FFA4  │00000000
0006FFA8  │00000000
0006FFAC  │7FFDF000
0006FFB0  │00000000
0006FFB4  │00000000
0006FFB8  │00000000
```

```
Analysing minesam: 77 heuristical procedures, 150 calls to known, 48 calls to guessed functions
```

From the OllyDbg menu bar, click **View**, **Memory**.

The memory segments are shown, as shown below.

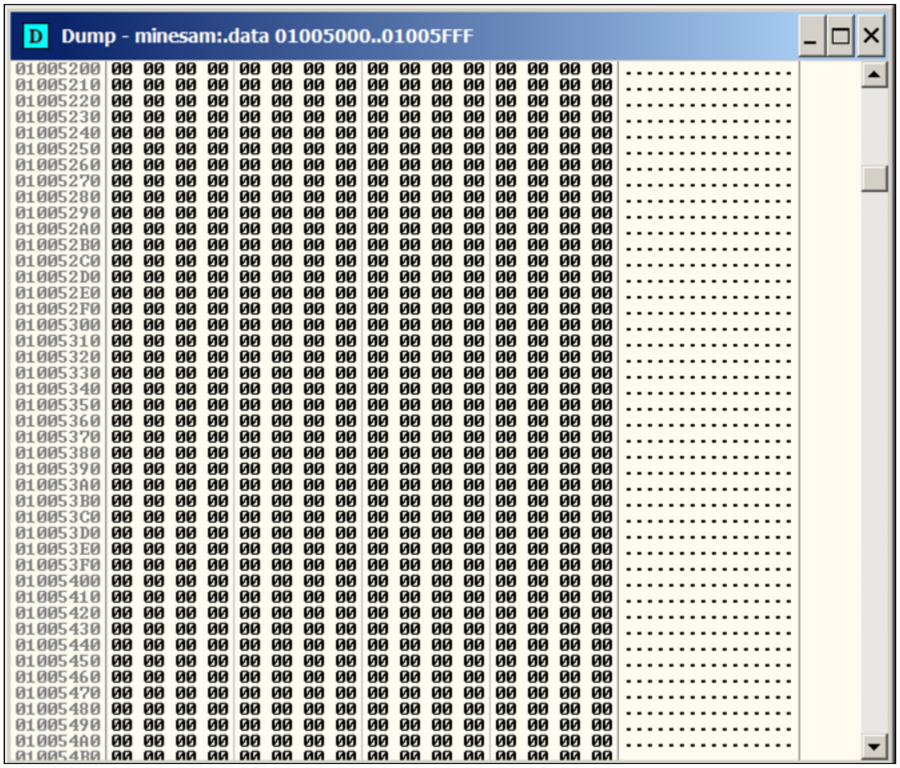
Right-click the **minesam.data** line and click **Dump**, as shown below.


OllyDbg - minesam.exe - [Memory map]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Actualize
Dump in CPU
Dump
Search Ctrl+B
Set break-on-access F2
Set memory breakpoint on access
Set memory breakpoint on write
Set access
Copy to clipboard
Sort by
Appearance

Address	Size	Owner	Section	Contains	Type	Access	Initial	Mapped as
00010000	00010000				Map	RW	RW	
00020000	00001000				Priv	RW	RW	
0006A000	00002000				Priv	RW Guar	RW	
0006C000	00004000			stack of ma:	Priv	RW Guar	RW	
00070000	00004000				Map	R	R	
00080000	00002000				Map	R	R	
00090000	00003000				Map	R	R	
00150000	00003000							
00160000	00001000							
00170000	00002000							
00180000	00002000							
00190000	00001000							\HarddiskVolume1\Windows\System32\oleaccrc.dl
001B0000	00002000							
001D0000	00003000							
00220000	0000A000							
00320000	00380000							\HarddiskVolume1\Windows\System32\loc2008.nls
006A0000	00103000							
00970000	00008000							
00980000	00380000							\HarddiskVolume1\Windows\System32\loc2008.nls
01000000	00001000	minesam		PE header				
01001000	00004000	minesam	.text	code,impor				
01005000	00001000	minesam	.data	data	Imag	R	RWE	
01006000	0001A000	minesam	.rsrc	resources	Imag	R	RWE	
01020000	0003B000				Map	R	R	

In the Dump window, scroll down to show memory near **01005340**.

This area contains only zeroes, as shown below.



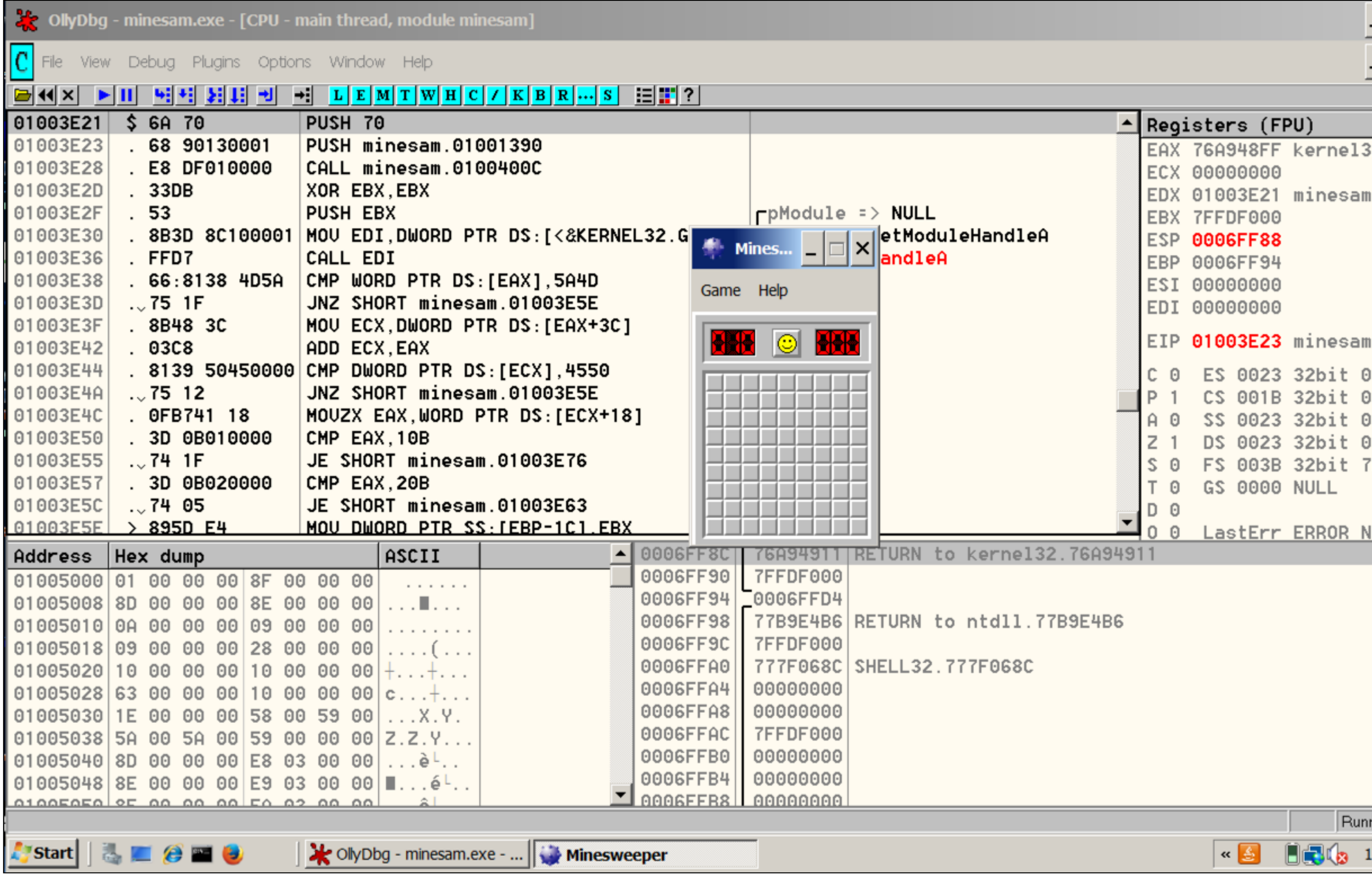
```
01005200 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005210 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005220 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010052F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005330 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005340 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005350 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005360 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005370 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005380 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005390 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010053F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005410 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005420 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005490 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010054A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010054B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

From the OllyDbg menu bar, click **View**, **CPU**.

From the OllyDbg menu bar, click **Debug**, **Run**.

If the lower-right corner of OllyDbg still shows a "Paused" message, click **Debug**, **Run** again.

A Minesweeper window opens, but does not come to the front. Click its button on the taskbar to bring it to the front, as shown below.



OllyDbg - minesam.exe - [CPU - main thread, module minesam]
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Mines...
Game Help
Address Hex dump ASCII
Registers (FPU)
Runn
Start
OllyDbg - minesam.exe - ...
Minesweeper

```
01003E21  $ 6A 70          PUSH 70
01003E23  . 68 90130001    PUSH minesam.01001390
01003E28  . E8 DF010000    CALL minesam.0100400C
01003E2D  . 33DB           XOR EBX,EBX
01003E2F  . 53             PUSH EBX                                ┌pModule => NULL
01003E30  . 8B3D 8C100001  MOV EDI,DWORD PTR DS:[<&KERNEL32.G      etModuleHandleA
01003E36  . FFD7           CALL EDI                                andleA
01003E38  . 66:8138 4D5A   CMP WORD PTR DS:[EAX],5A4D
01003E3D  .v75 1F          JNZ SHORT minesam.01003E5E
01003E3F  . 8B48 3C        MOV ECX,DWORD PTR DS:[EAX+3C]
01003E42  . 03C8           ADD ECX,EAX
01003E44  . 8139 50450000  CMP DWORD PTR DS:[ECX],4550
01003E4A  .v75 12          JNZ SHORT minesam.01003E5E
01003E4C  . 0FB741 18      MOVZX EAX,WORD PTR DS:[ECX+18]
01003E50  . 3D 0B010000    CMP EAX,10B
01003E55  .v74 1F          JE SHORT minesam.01003E76
01003E57  . 3D 0B020000    CMP EAX,20B
01003E5C  .v74 05          JE SHORT minesam.01003E63
01003E5E  > 895D E4        MOV DWORD PTR SS:[EBP-1C],EBX
```

```
EAX 76A948FF kernel3
ECX 00000000
EDX 01003E21 minesam
EBX 7FFDF000
ESP 0006FF88
EBP 0006FF94
ESI 00000000
EDI 00000000

EIP 01003E23 minesam

C 0  ES 0023 32bit 0
P 1  CS 001B 32bit 0
A 0  SS 0023 32bit 0
Z 1  DS 0023 32bit 0
S 0  FS 003B 32bit 7
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR N
```

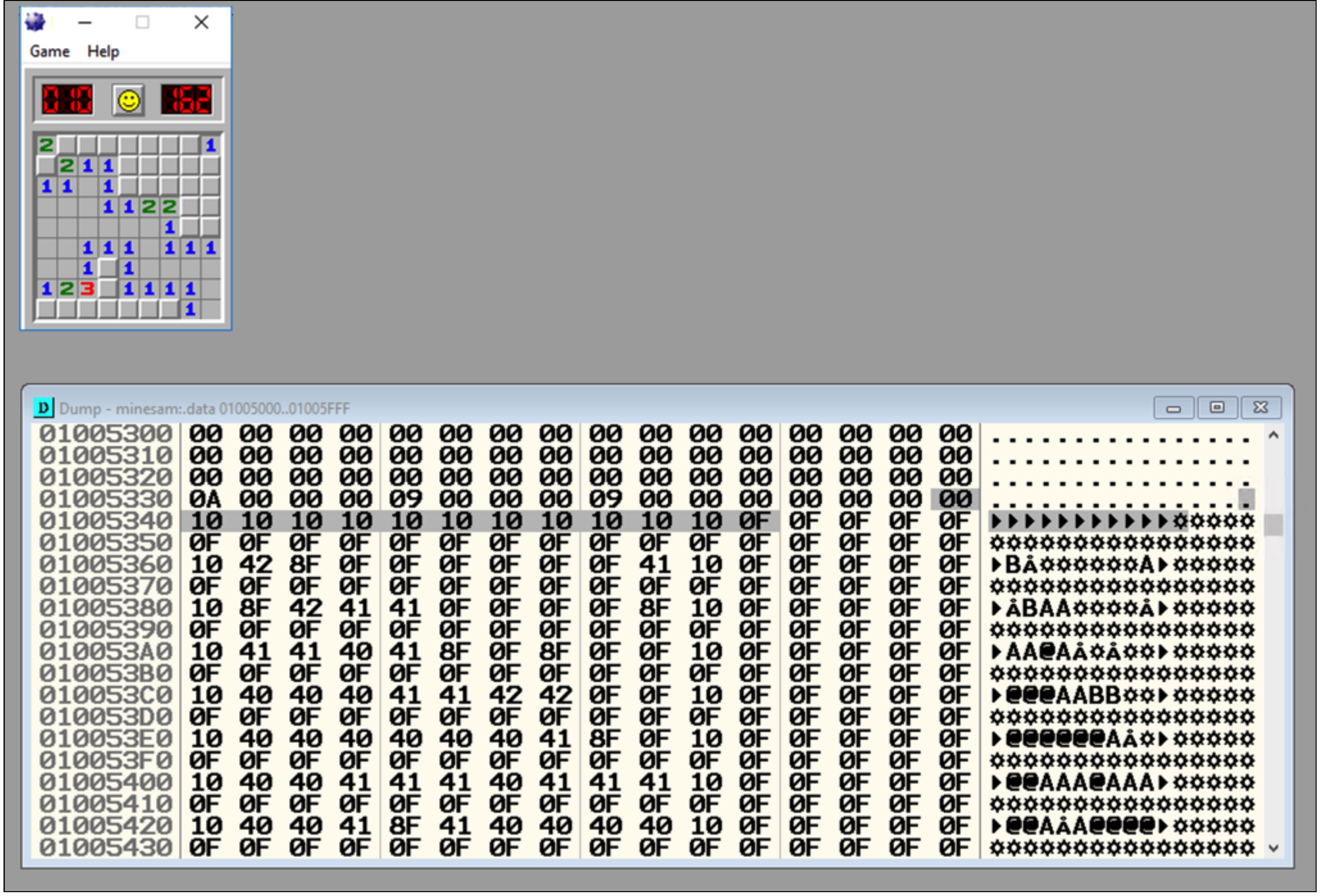
```
Address   Hex dump                 ASCII
01005000  01 00 00 00 8F 00 00 00   ......
01005008  8D 00 00 00 8E 00 00 00  ...█...
01005010  0A 00 00 00 09 00 00 00  ........
01005018  09 00 00 00 28 00 00 00  ....(...
01005020  10 00 00 00 10 00 00 00  +...+...
01005028  63 00 00 00 10 00 00 00  c...+...
01005030  1E 00 00 00 58 00 59 00  ...X.Y.
01005038  5A 00 5A 00 59 00 00 00  Z.Z.Y...
01005040  8D 00 00 00 E8 03 00 00  ...è└..
01005048  8E 00 00 00 E9 03 00 00  █...é└..
01005050  8F 00 00 00 EA 03 00 00     ê└
```

```
0006FF8C  76A94911  RETURN to kernel32.76A94911
0006FF90  7FFDF000
0006FF94  0006FFD4
0006FF98  77B9E4B6  RETURN to ntdll.77B9E4B6
0006FF9C  7FFDF000
0006FFA0  777F068C  SHELL32.777F068C
0006FFA4  00000000
0006FFA8  00000000
0006FFAC  7FFDF000
0006FFB0  00000000
0006FFB4  00000000
0006FFB8  00000000
```

In Minesweeper, click a cell to change the display.

From the OllyDbg menu bar, click **Window,Dump**.

Compare the Minesweeper gameboard with the Dump window. You can see that the gameboard is stored in RAM, using an "A" for "1", and a "B" for "2", as shown below.



Dump - minesam:.data 01005000..01005FFF

```
01005300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005330 0A 00 00 00 09 00 00 00 09 00 00 00 00 00 00 00 ................
01005340 10 10 10 10 10 10 10 10 10 10 10 0F 0F 0F 0F 0F ▸▸▸▸▸▸▸▸▸▸▸☼☼☼☼☼
01005350 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005360 10 42 8F 0F 0F 0F 0F 0F 0F 41 10 0F 0F 0F 0F 0F ▸BÅ☼☼☼☼☼☼A▸☼☼☼☼☼
01005370 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005380 10 8F 42 41 41 0F 0F 0F 0F 8F 10 0F 0F 0F 0F 0F ▸ÅBAA☼☼☼☼Å▸☼☼☼☼☼
01005390 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053A0 10 41 41 40 41 8F 0F 8F 0F 0F 10 0F 0F 0F 0F 0F ▸AA@AÅ☼Å☼☼▸☼☼☼☼☼
010053B0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053C0 10 40 40 40 41 41 42 42 0F 0F 10 0F 0F 0F 0F 0F ▸@@@AABB☼☼▸☼☼☼☼☼
010053D0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053E0 10 40 40 40 40 40 40 41 8F 0F 10 0F 0F 0F 0F 0F ▸@@@@@@AÅ☼▸☼☼☼☼☼
010053F0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005400 10 40 40 41 41 41 40 41 41 41 10 0F 0F 0F 0F 0F ▸@@AAA@AAA▸☼☼☼☼☼
01005410 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005420 10 40 40 41 8F 41 40 40 40 40 10 0F 0F 0F 0F 0F ▸@@AÅA@@@@▸☼☼☼☼☼
01005430 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
```

If we can read the RAM, we can cheat at the game.

Notice the highlighted region in the image above. If we can find this sequence of bytes in RAM, we can find the gameboard in a memory dump.

Getting Procdump

In a Web browser, go to

https://docs.microsoft.com/en-us/sysinternals/downloads/procdump

Download Procdump.zip, and put it in your Downloads folder.

Click **Start**, **Computer**. Navigate to your Download folder.

Right-click **Procdump.zip** and click "**Extract All...**", **Extract**.

Creating a Python Script

We can automate the process with Python.

Click **Start**. Type **CMD**. Open a Command Prompt window, and execute these commands:

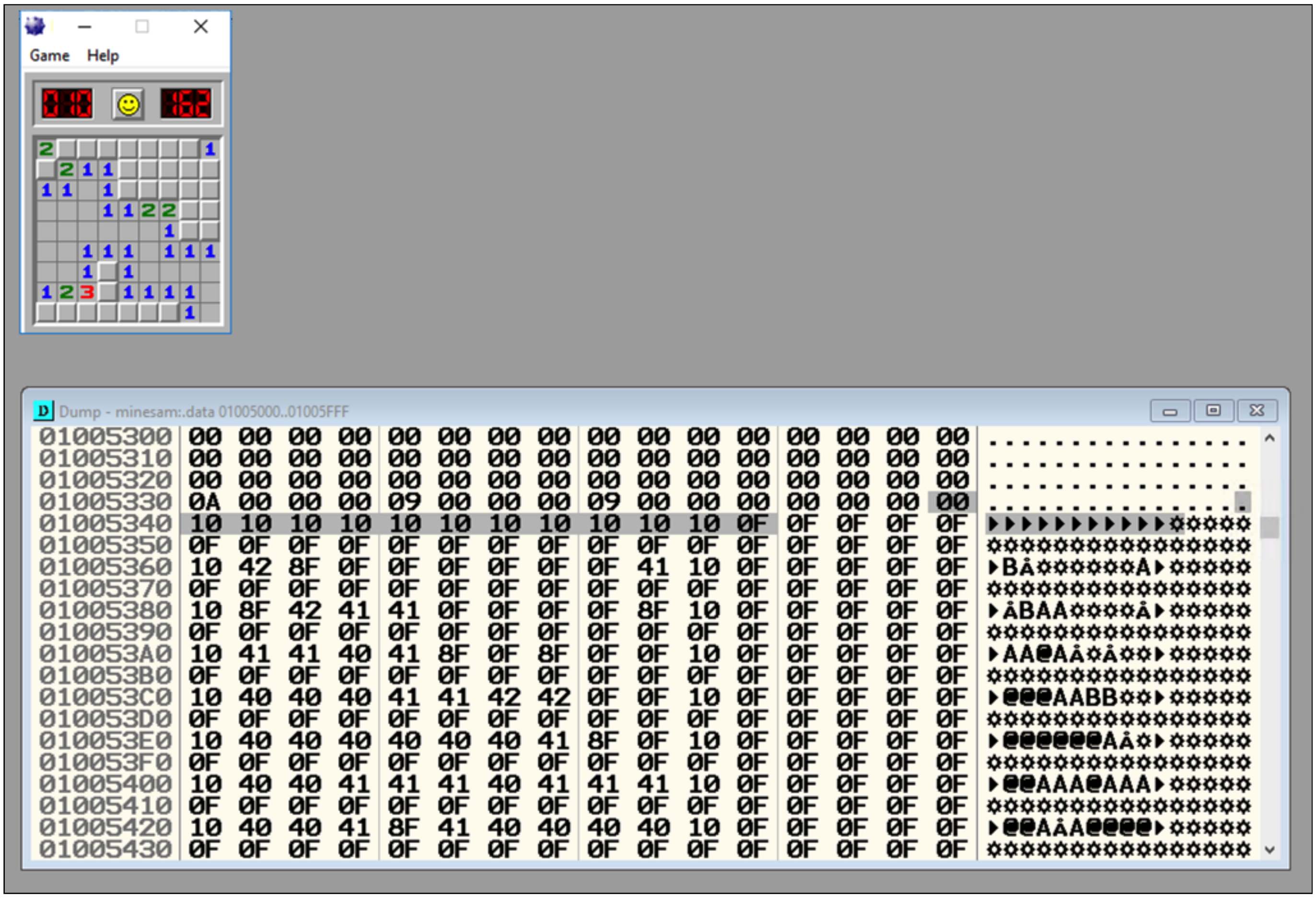
```
cd Downloads\procdump
notepad cheat.py
```

If a license agreement pops up, agree to it.

A box pops up, saying "Do you want to create a ne file...?". Click **Yes**.

Paste in this code, as shown below.

```
import os

# Dump memory

cmd = "del mine.dmp"
os.system(cmd)
cmd = "procdump -ma minesam.exe mine"
os.system(cmd)

# Find gameboard

mark ='\x00\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x10\x0F'

line_length = 32
board_size = 500 # characters in whole board

with open("mine.dmp", "rb") as f:
  data= f.read()

start = data.find(mark)
if start <0:
  print "Gameboard not found"

# Print gameboard

for i in range(0, board_size, line_length):
  line = ''
  for j in range(line_length):
    g = data[start+i+j]
    if g == '\x10':
      c = "-"
    elif g == '\x0f':
      c = " "
    elif g == '\x8f':
      c = "*"
    elif g == '\x00':
      c = " "
    else:
      c = chr( ord(g) - 16 )
    line += c
  print line
```



Dump - minesam:.data 01005000..01005FFF

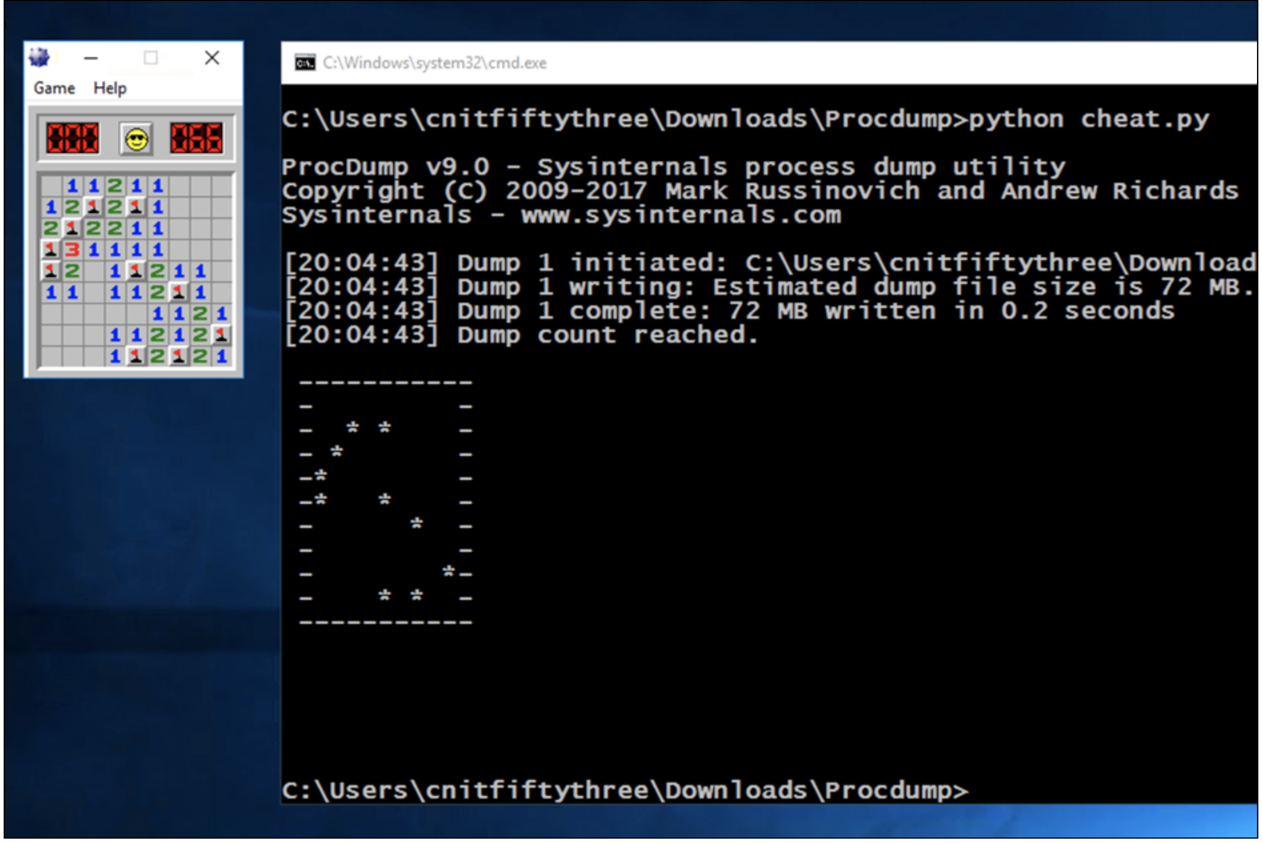
```
01005300 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005310 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005320 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
01005330 0A 00 00 00 09 00 00 00 09 00 00 00 00 00 00 00 ................
01005340 10 10 10 10 10 10 10 10 10 10 10 0F 0F 0F 0F 0F ▶▶▶▶▶▶▶▶▶▶▶☼☼☼☼☼
01005350 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005360 10 42 8F 0F 0F 0F 0F 0F 0F 41 10 0F 0F 0F 0F 0F ▶BÅ☼☼☼☼☼☼A▶☼☼☼☼☼
01005370 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005380 10 8F 42 41 41 0F 0F 0F 0F 8F 10 0F 0F 0F 0F 0F ▶ÅBAA☼☼☼☼Å▶☼☼☼☼☼
01005390 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053A0 10 41 41 40 41 8F 0F 8F 0F 0F 10 0F 0F 0F 0F 0F ▶AA@AÅ☼Å☼☼▶☼☼☼☼☼
010053B0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053C0 10 40 40 40 41 41 42 42 0F 0F 10 0F 0F 0F 0F 0F ▶@@@AABB☼☼▶☼☼☼☼☼
010053D0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
010053E0 10 40 40 40 40 40 40 41 8F 0F 10 0F 0F 0F 0F 0F ▶@@@@@@AÅ☼▶☼☼☼☼☼
010053F0 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005400 10 40 40 41 41 41 40 41 41 41 10 0F 0F 0F 0F 0F ▶@@AAA@AAA▶☼☼☼☼☼
01005410 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
01005420 10 40 40 41 8F 41 40 40 40 40 10 0F 0F 0F 0F 0F ▶@@AÅA@@@@▶☼☼☼☼☼
01005430 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼☼
```

In the Notepad window, click **File**, **Save**.

In the Command Prompt window, execute this command:

```
python cheat.py
```

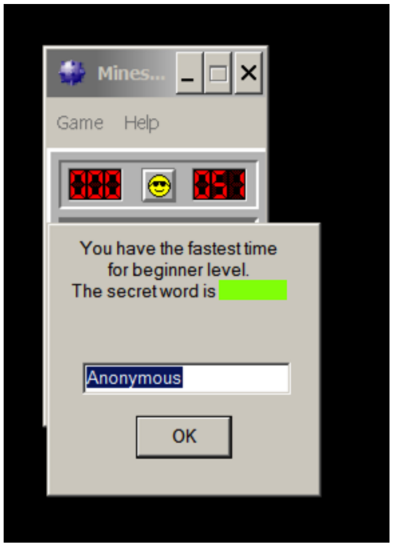
The program shows the location of the mines. With this information, you should easily be able to click all the squares without mines, as shown below.



```
C:\Windows\system32\cmd.exe

C:\Users\cnitfiftythree\Downloads\Procdump>python cheat.py

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[20:04:43] Dump 1 initiated: C:\Users\cnitfiftythree\Download
[20:04:43] Dump 1 writing: Estimated dump file size is 72 MB.
[20:04:43] Dump 1 complete: 72 MB written in 0.2 seconds
[20:04:43] Dump count reached.

 -----------
 -         -
 -  * *    -
 - *       -
 -*        -
 -*   *    -
 -      *  -
 -         -
 -        *-
 -    * *  -
 -----------

C:\Users\cnitfiftythree\Downloads\Procdump>
```

Flag PMA 402.1: Beginner Level (15 pts)

When you win the game, a secret word will appear, which is covered by a green box in the image below. That's the flag.

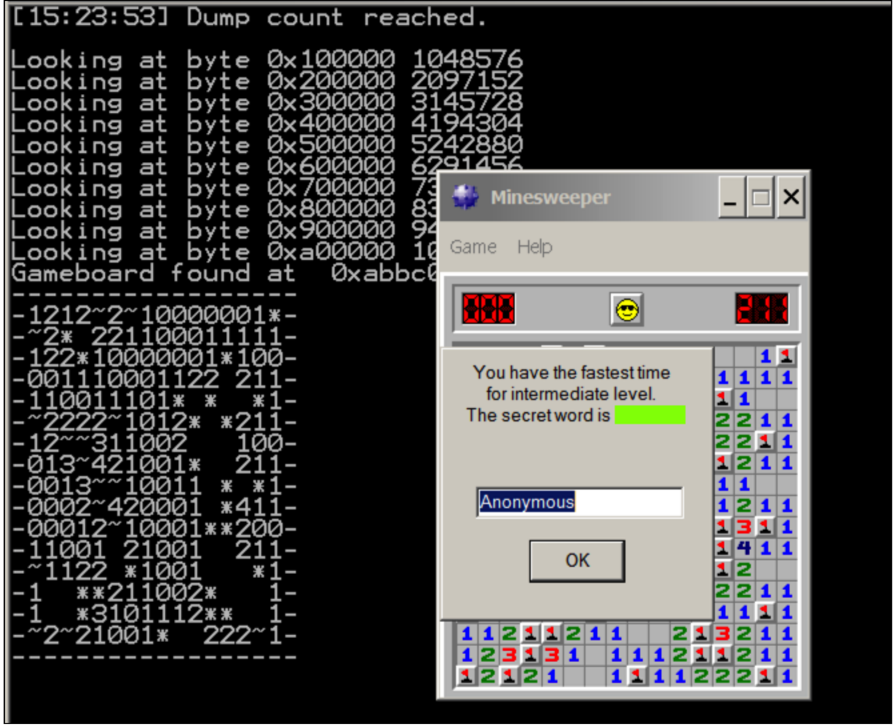


You have the fastest time for beginner level.
The secret word is [redacted]

Flag PMA 402.2: Intermediate Level

In Minesweeper, click **Game**, **Intermediate**.

Create a cheating tool that works for this level and win the game, as shown below.



```
[15:23:53] Dump count reached.

Looking at byte 0x100000 1048576
Looking at byte 0x200000 2097152
Looking at byte 0x300000 3145728
Looking at byte 0x400000 4194304
Looking at byte 0x500000 5242880
Looking at byte 0x600000 6291456
Looking at byte 0x700000 73
Looking at byte 0x800000 83
Looking at byte 0x900000 94
Looking at byte 0xa00000 10
Gameboard found at  0xabbc0
------------------
-1212~2~10000001*-
-~2* 221100011111-
-122*10000001*100-
-001110001122 211-
-110011101* *  *1-
-~2222~1012* *211-
-12~~311002   100-
-013~421001*  211-
-0013~~10011 * *1-
-0002~420001 *411-
-00012~10001**200-
-11001 21001  211-
-~1122 *1001   *1-
-1  **211002*   1-
-1  *3101112**  1-
-~2~21001*  222~1-
------------------
```

You have the fastest time
for intermediate level.
The secret word is [redacted]

Anonymous

Flag PMA 402.3: Expert Level (10 pts extra)

In Minesweeper, click **Game**, **Expert**.

Find the secret word for the Expert level.

Hint: use a totally different technique; don't play the game.

Sources

[Game Hacking: WinXP Minesweeper](#)
[_MINIDUMP_TYPE Enumeration](#)

Posted 9-18-18
Revised for Win 2016 9-11-19
OllyDbg download link fixed 10-1-20
Updated in minor ways 2-23-21