What You Need

A Windows machine, real or virtual. I used a Windows Server 2008 virtual machine.

Purpose

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

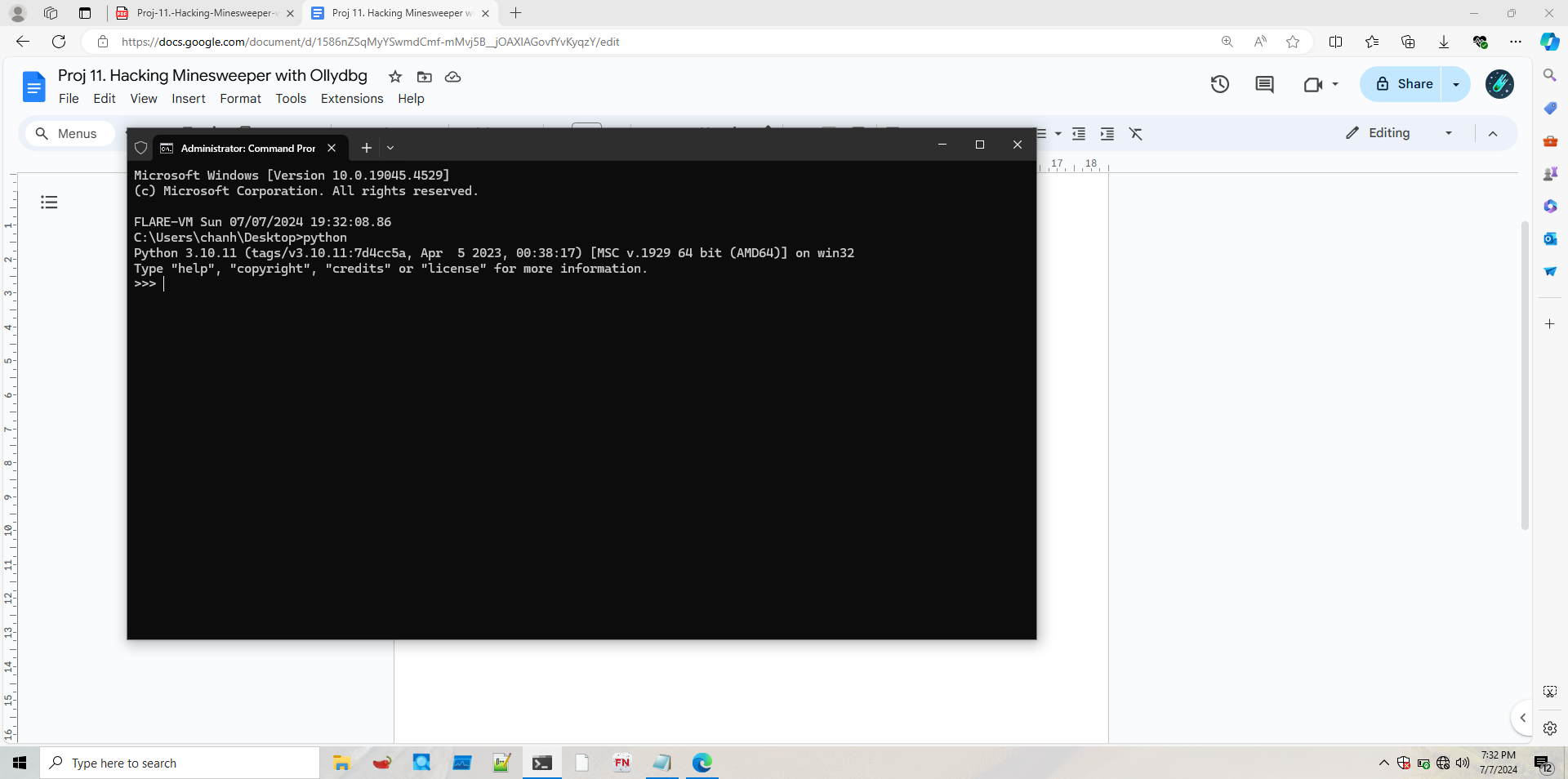
Getting Python

The Windows Server 2008 machine we have been using already has Python installed.

To see if you have it, open a Command Prompt and execute this command:

python

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



Getting Minesweeper

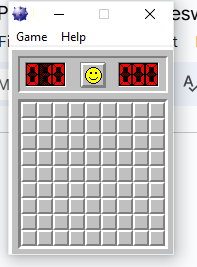
Download the minesweeper program from the link below.

minesam.exe.zip

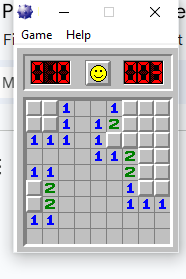
Right-click the zipped file and click "Extract All...", Extract.

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

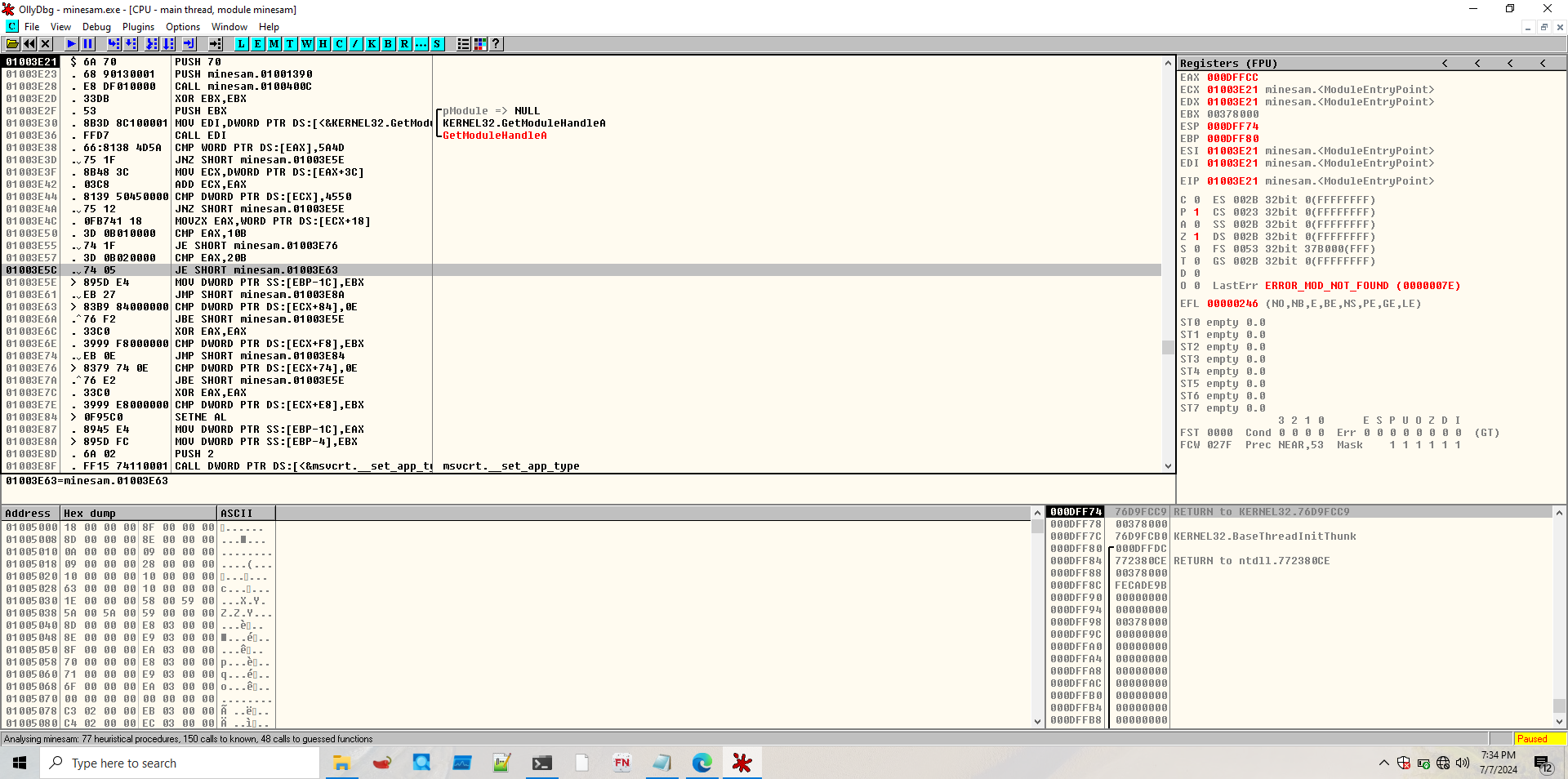
.

Viewing the Game in OllyDbg

Close Minesweeper.

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

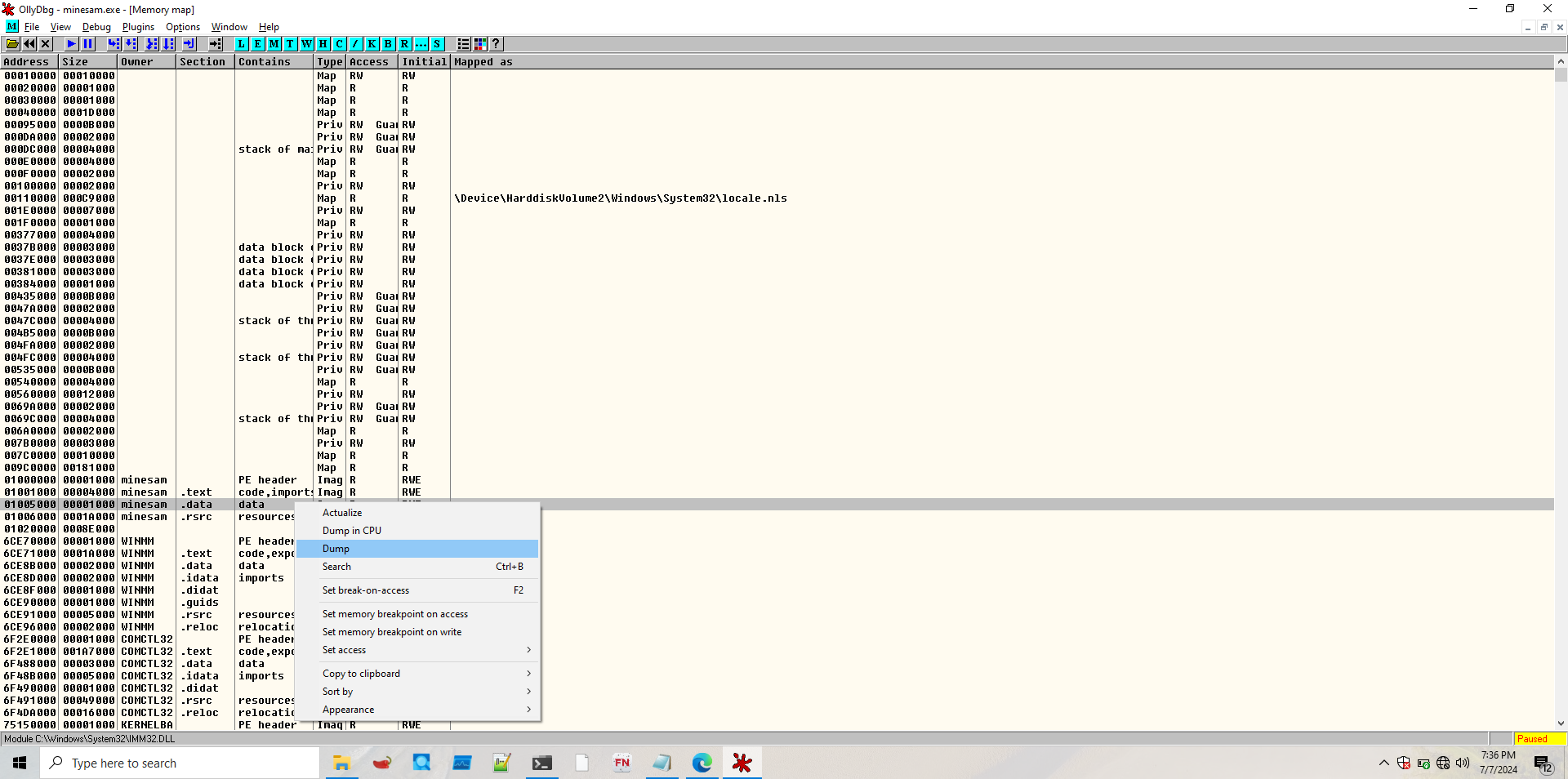
The program loads and pauses, as shown below



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



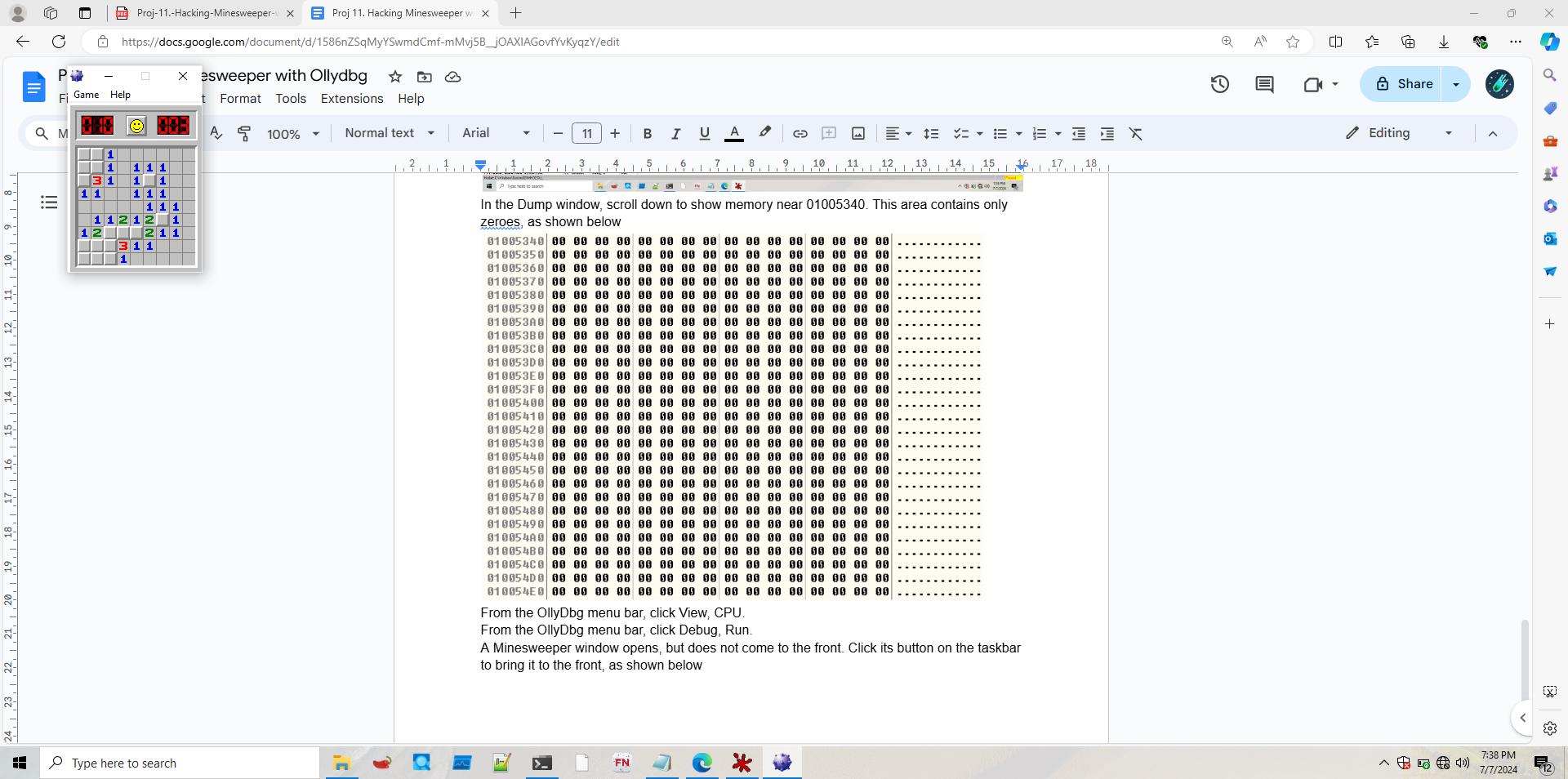
In the Dump window, scroll down to show memory near 01005340. This area contains only zeroes, as shown below



From the OllyDbg menu bar, click View, CPU.

From the OllyDbg menu bar, click Debug, Run.

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



Viewing the Stored Gameboard

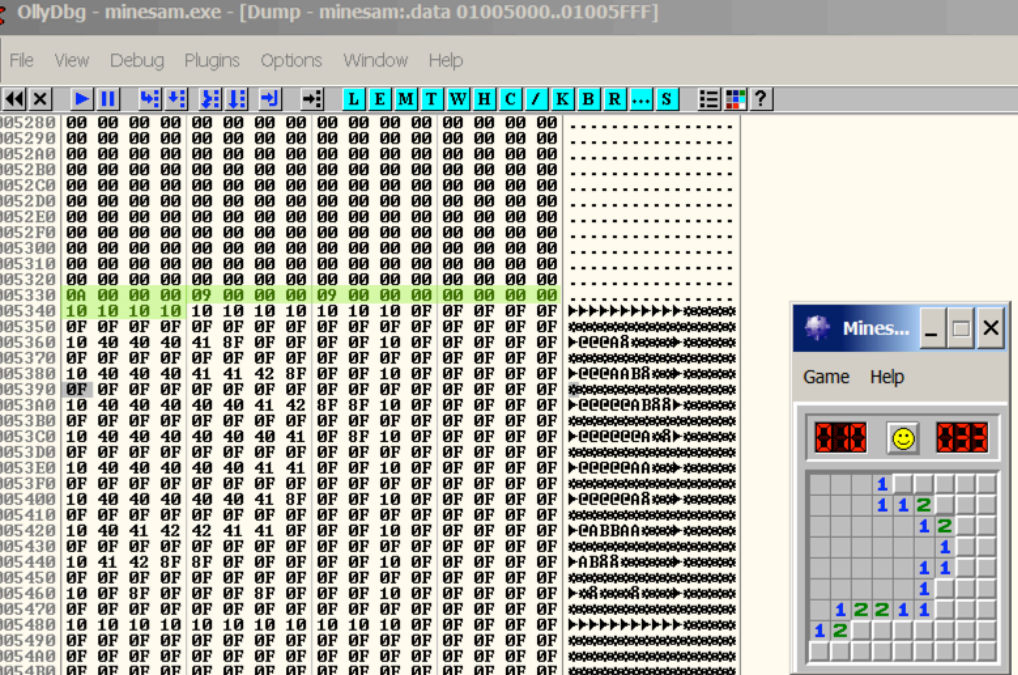
From the OllyDbg menu bar, click Window, Dump.

The memory after 01005340 now contains data, as shown below



Click the Minesweeper button on the taskbar to bring it to the front. Click a cell to change the display.

Comare 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



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.

Capturing Process Memory

Close Minesweeper. Close OllyDbg. Double-click minesam.exe to run Minesweeper again.

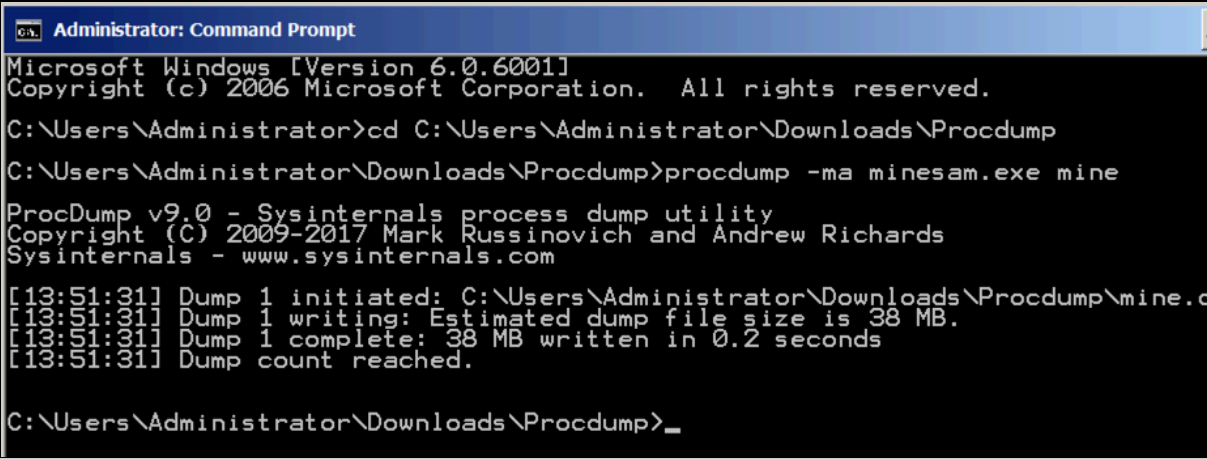
Open a Command Prompt and execute these commands:

cd C:\Users\Administrator\Downloads\Procdump

procdump -ma minesam.exe mine

A box pops up, titled ProcDump License Agreement. Click Agree.

Procdump makes a dump file, as shown below



Viewing the Memory with HxD

The Windows Server 2008 machine we have been using already has HxD installed.

If you don't have it, get it here:

https://mh-nexus.de/en/hxd/

Open HxD. From the HxD menu bar, click File, Open.

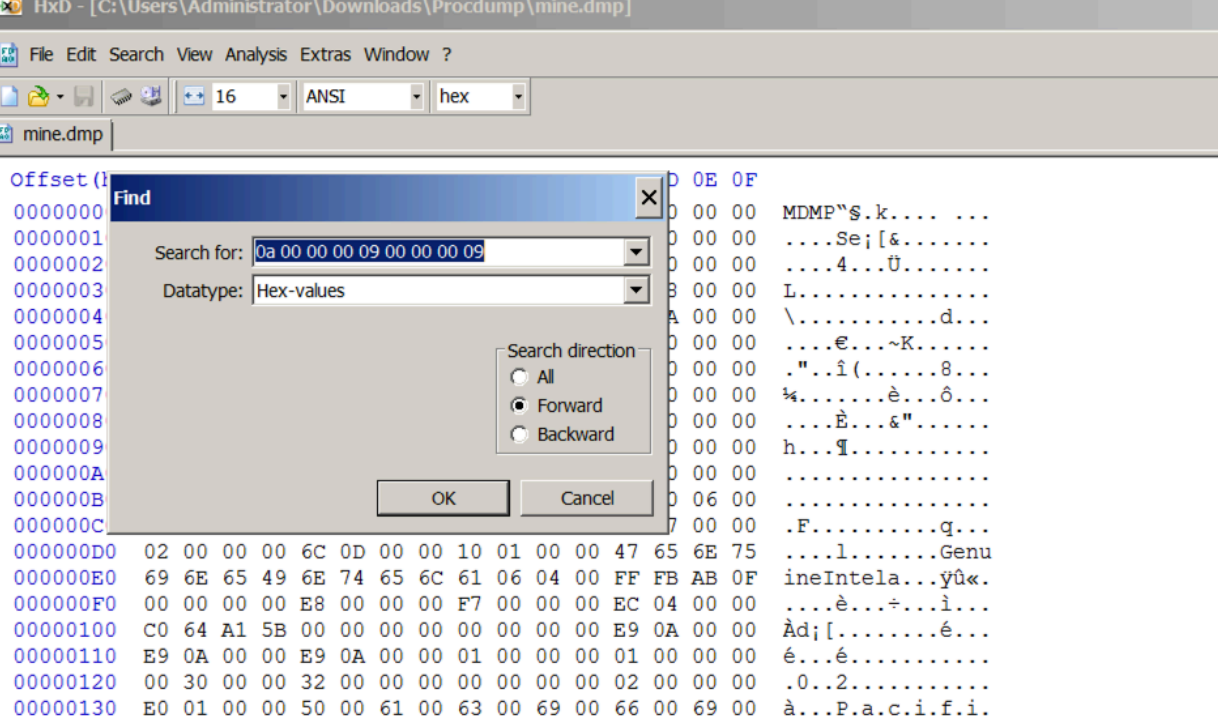
Navigate to your Downloads folder. Open the Procdump folder and double-click the mine.dmp file.

From the HxD menu bar, click Search, Find.

In the "Find" field, select a Datatype of Hex-values.

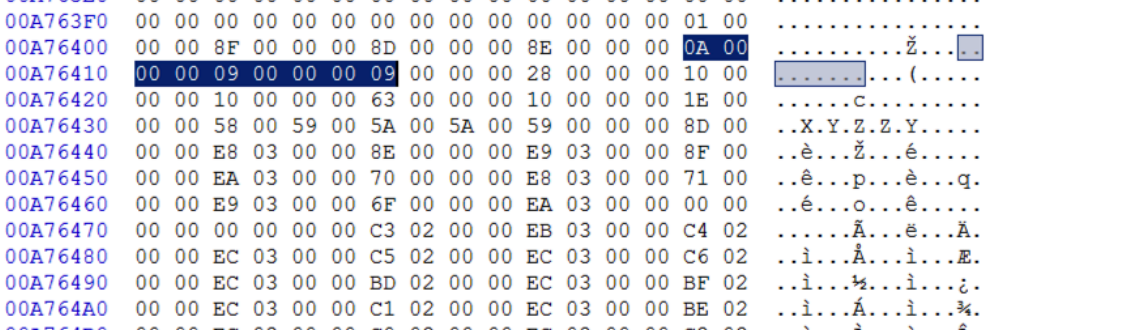
In the "Search for" field, enter this text, as shown below.

0a 00 00 00 09 00 00 00 09



In the "Find" box, click OK.

The string is found, but it may not be the correct hit. The first one doesn't have the gameboard after it, as shown below



Creating a Python Script

We can automate the process with Python. In a Command Prompt window, execute these commands:

cd C:\Users\Administrator\Downloads\Procdump

notepad cheat.py

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 ='\x0A\x00\x00\x00\x09\x00\x00\x00\x09\x00\x00\x00\x00\x00\x00\x00\x10\x10\x10\x10'

nread = 20

boardfound = 0

gameboard = []

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

line = f.read(20)

while (boardfound == 0):

c = f.read(1)

if c == "":

print "File ended, but gameboard not found!"

exit()

line = line[1:] + c

nread += 1

if nread % 0x100000 == 0:

print "Looking at byte", hex(nread), nread

if line == mark:

print "Gameboard found at ", hex(nread)

boardfound = 1

for i in range(4):

gameboard.append('\x10')

for i in range(500):

gameboard.append(f.read(1))

# Print Gameboard

l = len(gameboard)

m = 32 # items per line

for i in range(0, l-m, m):

line = ""

for j in range(m):

g = gameboard[i+j]

# print i, j, ord(g)

if g == '\x10':

c = "-"

elif g == '\x0f':

c = " "

elif g == '\x8f':

c = "\*"

elif g == '\x00':

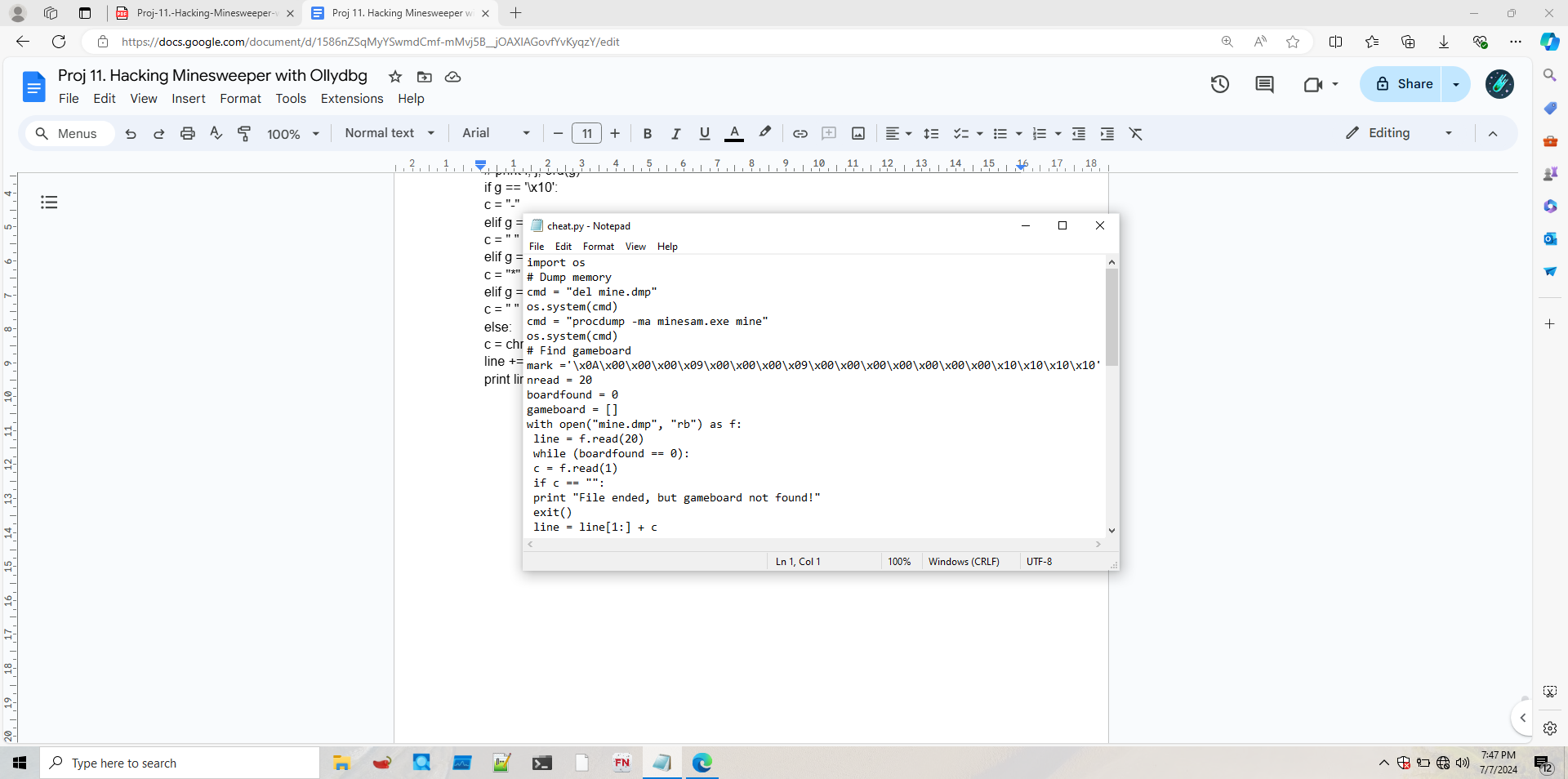
c = " "

else:

c = chr( ord(g) - 16 )

line += c

print line

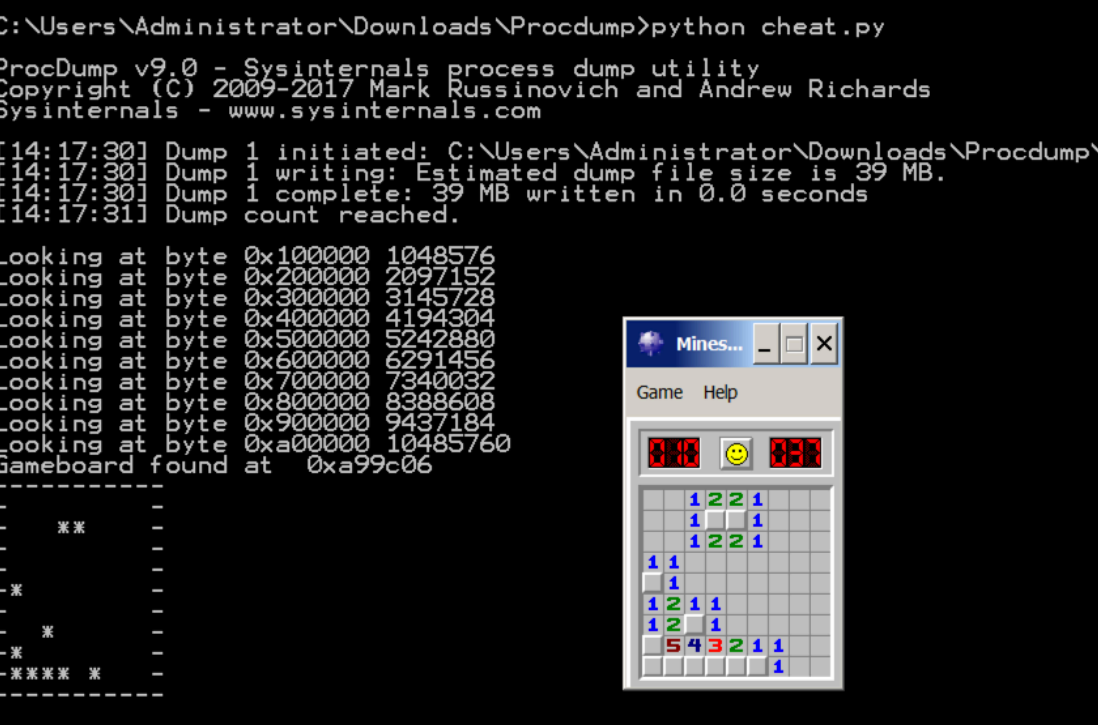


In the Notepad window, click File, Save.

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



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

