Malware Analysis
Fall 2015 — Project 2

Maxfield Chen

November 3, 2015

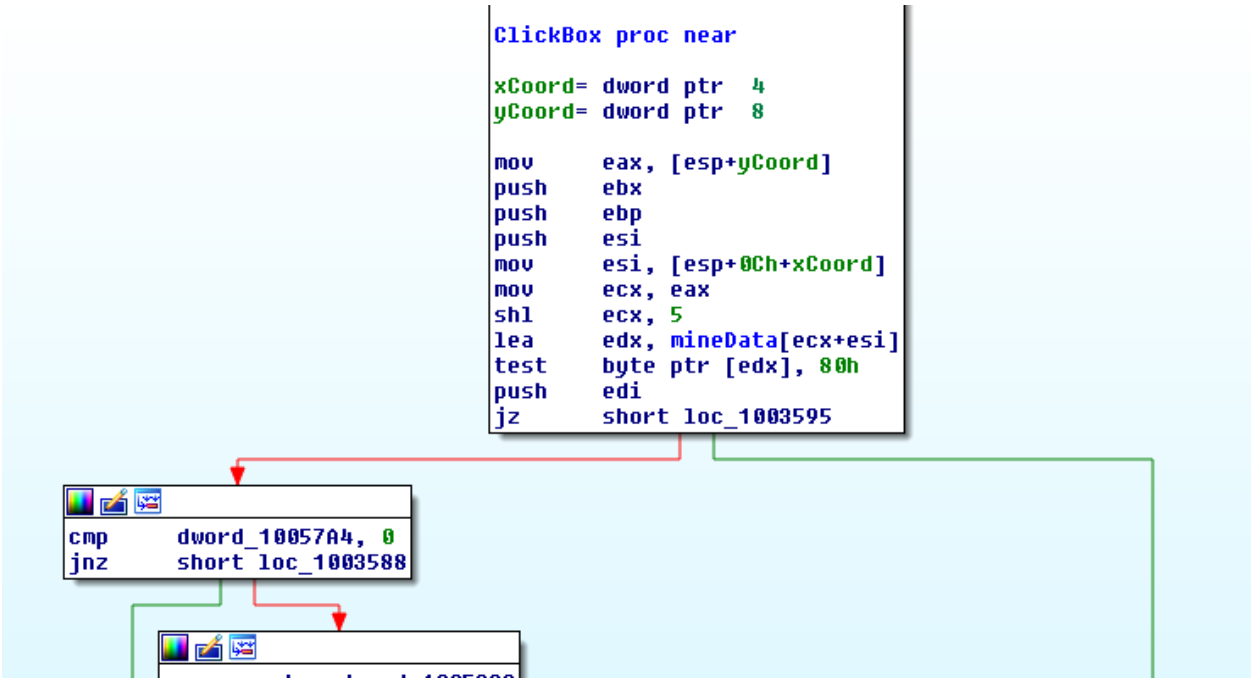
Advanced Analysis

1. Purpose: I needed to identify several key functions and variables in order to begin formulating a trainer. I specifically looked for timer variables, win functions, and sought to find the board representation.

2. AUTO WIN:

   (a) I started by poking through IDA functions and through some luck, stumbled onto the function called during a click. This was made evident by the two integer arguments it took and the structure of the function.

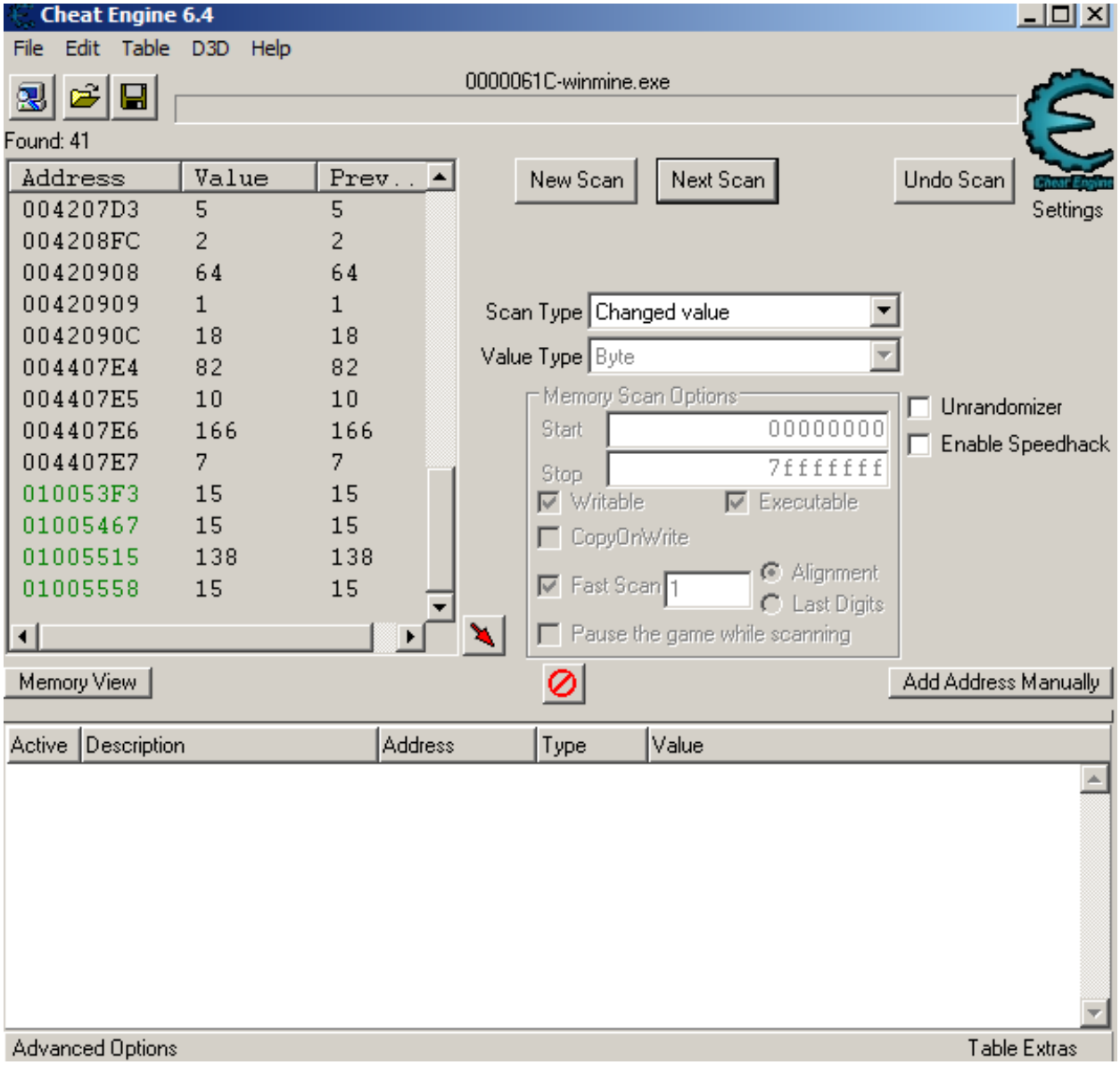
   (b)



   (c) Having this, all I needed was to be able to detect if a certain square was a mine. I downloaded cheatengine and used it to scan the memory loaded by Minesweeper.

   (d) I didn't know exactly where the mines were stored so I scanned once getting all bytes.

(e) I changed the board and made another scan for changed values. Repeating this a couple times and poking through IDA to confirm my findings, I was able to get the address of the array. The values found in the array suggested that 142 and 143 are mine squares.



(f)

(g) From there it was a simple matter to loop through the board and call click on all squares which were not mines!

3. Show Mines:

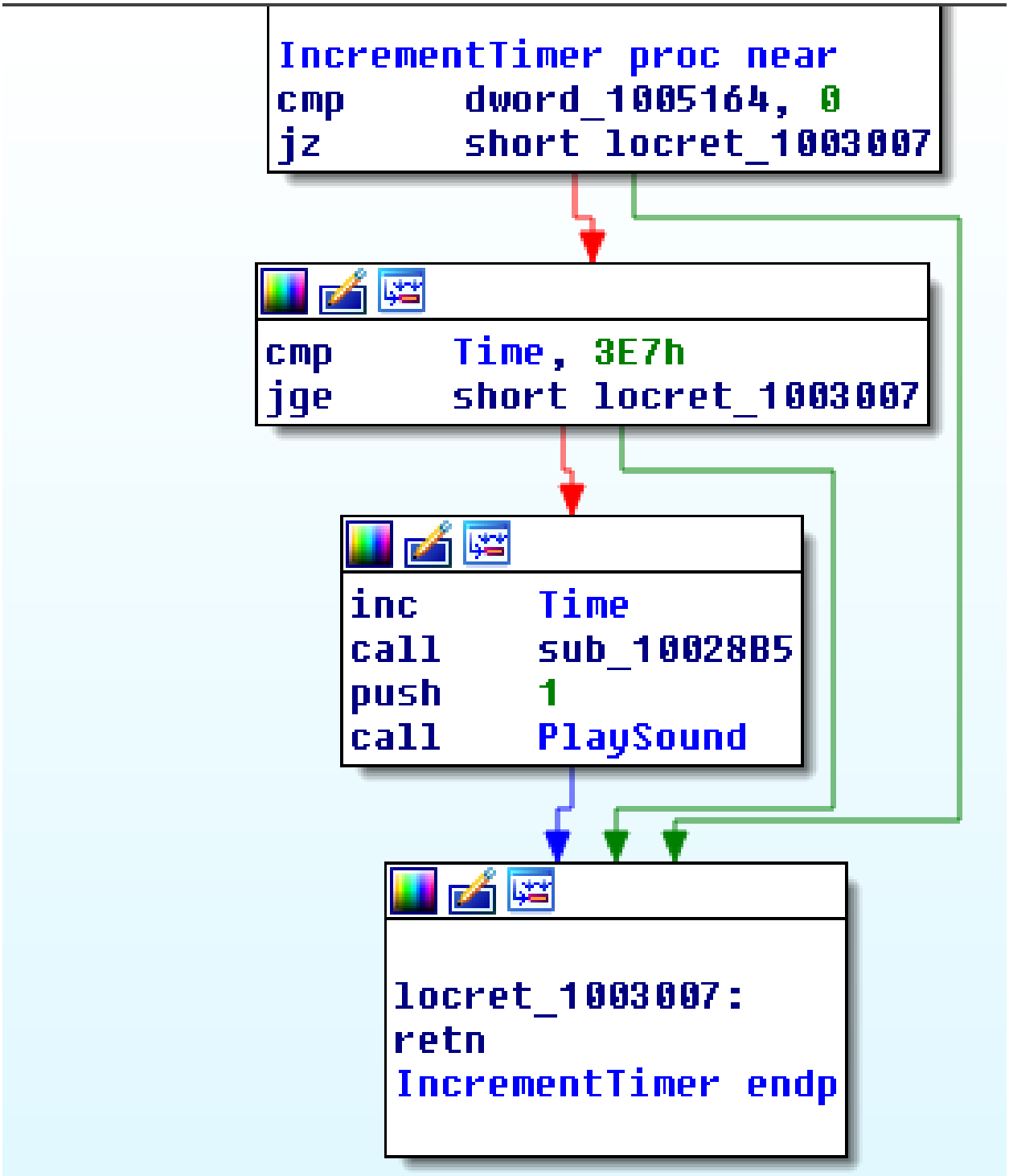
(a) Looking through the click function I had my next breakthrough. I found a function calling getDC, and after some google-fu it become clear that this function was used to draw graphics on various squares. Nifty!

```
; int __stdcall reDrawSquare(int xCoord, int yCoord, char a3)
reDrawSquare proc near

xCoord= dword ptr  4
yCoord= dword ptr  8
arg_8= byte ptr  0Ch

mov     eax, [esp+yCoord]
mov     ecx, [esp+xCoord]
push    [esp+yCoord]
shl     eax, 5
lea     eax, mineData[eax+ecx]
mov     dl, [eax]
and     dl, 0E0h
or      dl, [esp+4+arg_8]
push    ecx
mov     [eax], dl
call    UpdateSquareImage
retn    0Ch
reDrawSquare endp
```

(b)

(c) Now that I had a method to draw on squares I just needed to figure out what argument provided an acceptable drawing. I chose the ? and found it through more cheatengine analysis and guessing.

4. Freeze Timer:



(a)

(b) A little more poking around and I found this highly suggestive function which I correctly assumed was the timer increment function. Seems hookable.

(c) I downloaded, built and linked the detours library. After looking at the previous lab and other examples I got a hook set on the previously found timer function.

(d) This hook simply nops freezing the timer.

5. Inert Mines:

   (a) I took a very similar approach to the previous problem, I simply hooked the click function and if the square was a mine drew a ? instead of calling the click function.

   (b) If the square was not a mine the trampoline was called and the real click function is executed.

6. Extract Layout:

   (a) Surprisingly this gave me the most trouble of all the tasks. For some reason I simply could not get the array to play nicely and my attempts to calculate the hidden squares were resulting in silent failures.

   (b) After a few hours of horrifying debug statements using messagebox I decided that the prompt never said anything about side effects and having already built the autowin function, simply solved the board and read the resulting tiles from memory.

   (c) This was much easier.