

Reverse Engineering Course

Dynamic Analysis

Barak Gonen



Dynamic Analysis

- Easier to understand code functionality
 - Check values in registers, memory
 - Check stack: arguments and return addresses
- Modify code on the fly
- We shall learn:
 - WinDbg
 - x32dbg

TeaParty

- <https://data.cyber.org.il/reversing/TeaParty.zip>
- Copy both exe and pdb file to dedicated library





- Load TeaParty and list all modules – “lm”

```
0:000> lm
start      end          module name
00420000 00427000    TeaParty C (private pdb symbols)
54170000 54183000    VCRUNTIME140 (deferred)
5fee0000 5ff8f000    atcuf32 (deferred)
74c30000 74d4f000    ucrtbase (deferred)
76cc0000 76ebe000    KERNELBASE (deferred)
770c0000 771a0000    KERNEL32 (deferred)
77310000 774aa000    ntdll (pdb symbols)
```



- If debug info is present, any function can be searched
- X moduleName!*anything*
 - * is wildcard

```
0:000> x teaparty!main
00421080      TeaParty!main (void)
```

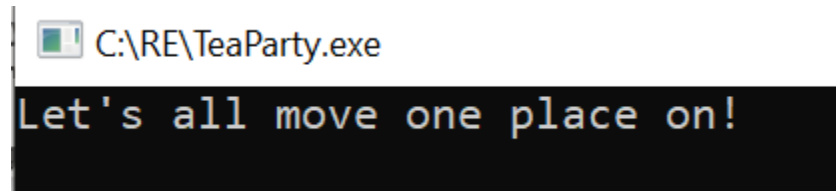
- Too Easy 😊

Finding main with the stack

- Step 1: take whatever program prints and search
- Step 2: any print should be called from main (if we go back enough)
- Step 3: look at the stack- return addresses

Find main, again

- Delete pdb file
 - C:\symbols\TeaParty
- Run program and check output



```
C:\RE\TeaParty.exe  
Let's all move one place on!
```

Search for Strings

- <https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/s--search-memory->
- S -a : search for **ascii** string
 - S -a startAddress endAddress string
 - Or : S -a startAddress L?HowManyBytesForward string
 - S -sa startAddress L:endAddress
 - Will search for any ascii string
 - -su for unicode

Display Memory

- da – Display Ascii
- da 423018
- Play with it: db, dw, dd

Start

- bp \$exentry
- g
- ba – BreakPoint Access
 - r – read
 - Number – memory size
- bl – List BreakPoints

```
0:000> bp $exentry
0:000> g
Breakpoint 0 hit
eax=00effc40 ebx=00d6b000
eip=004212f7 esp=00effbe8
cs=0023  ss=002b  ds=002b
TeaParty+0x12f7:
004212f7 e893030000
0:000> ba r1 423018
```

View the Stack

- g – run (to breakpoint)
- kb – view stack

```
0:000> kb
# ChildEBP RetAddr  Args to Child
00 00eff618 74c6a655 74d409fc 74d400e0 00d6b000 ucrtbase!__crt_stdio_output::output
01 00effabc 74c6a71f 3e3cfa39 74d409fc 74d400e0 ucrtbase!<lambda_0be4ab1c2a6918fda
02 00effaf0 74c6a7d7 00effb24 00effb10 00effb28 ucrtbase!__crt_seh_guarded_call<ir
03 00effb34 74c6a77c 00000004 00000000 00000000 ucrtbase!common_vfprintf<__crt_stc
04 00effb4c 00421035 00000004 00000000 74d3f318 ucrtbase!__stdio_common_vfprintf+0
WARNING: Stack unwind information not available. Following frames may be wrong.
05 00effb6c 00421067 74d3f318 00423018 00000000 TeaParty+0x1035
06 00effb8c 00421096 00423018 00effbe4 0042127a TeaParty+0x1067
07 00effb98 0042127a 00000001 01145210 01146550 TeaParty+0x1096
08 00effbe4 770d6359 00d6b000 770d6340 00effc50 TeaParty+0x127a
09 00effbf4 77377c24 00d6b000 13cfba69 00000000 KERNEL32!BaseThreadInitThunk+0x19
0a 00effc50 77377bf4 ffffffff 77398fdd 00000000 ntdll!__RtlUserThreadStart+0x2f
0b 00effc60 00000000 004212f7 00d6b000 00000000 ntdll!_RtlUserThreadStart+0x1b
```

Main

- Copy address to disassembly window

Disassembly
×

Address: 421096
☒ Follow current instruction

0042107e	cc	int	3
0042107f	cc	int	3
00421080	55	push	ebp
00421081	8bec	mov	ebp, esp
00421083	b801000000	mov	eax, 1
00421088	85c0	test	eax, eax
0042108a	740f	je	TeaParty+0x109b (0042109b)
0042108c	6818304200	push	offset TeaParty+0x3018 (00423018)
00421091	e8aaffffff	call	TeaParty+0x1040 (00421040)
00421096	83c404	add	esp, 4
00421099	eb0d	jmp	TeaParty+0x10a8 (004210a8)
0042109b	6838304200	push	offset TeaParty+0x3038 (00423038)
004210a0	e89bffff	call	TeaParty+0x1040 (00421040)
004210a5	83c404	add	esp, 4
004210a8	33c0	xor	eax, eax
004210aa	5d	pop	ebp
004210ab	c3	ret	

Go Up

- If we are deep inside the stack:
- gu – Go Up

r

- Any register can be shown
 - r eax
 - r eip
- ... Or set
 - r eax = 1
 - r eip = ... 😊

a, u

- Any instruction can be modified
- a address
 - Then, write assembly instruction
 - Press extra enter to exit
- View memory
 - u address
 - Or simply disassembly window

Memory Window

- Tip: easy to track any variable

Memory

Address:

00000000~	00EFFBA0	01	00	00	00	10	52	14	01
00000000~	00EFFBD0	00	00	00	00	40	FC	EF	00

Exercise

- Do it all over again
- Find several ways to modify main so it will let us drink tea 😊

BreakMe

- Now crack BreakMe using only WinDbg

Simple_WinDbg_ex

- BreakMe was Easy, right?
- Crack Simple_WinDbg_ex.exe
- https://data.cyber.org.il/reversing/simple_windbg_ex.exe

Homework

- <https://data.cyber.org.il/reversing/Secret.zip>