Introduction to binary exploitation

Information Systems Security @ II UWR 2019

github.com/mzr/intro_to_binary_exploitation

Ad!

- WTF is CTF?
- justCatTheFish team
  - 3rd in Poland
  - 20th worldwide
  - pwndbg
  - pwntools

What is binary security / exploitation ?

“Binary exploitation is the process of manipulating a compiled application such that it violates some trust boundary in a way that is advantageous to you, the attacker.”

“Everyone uses [Python, Java, C#, …] nowadays”

Binary code is still relevant!

↓

JITs, performance, OS constructs, etc.

Vuln ⊊ bug

Gynvael Coldwind @ Programistok 2016

- integer overflow
- type mismatch
- buffer overflow
- use after free
- double free
- format string vulnerability
- and many many other

Bug vs Vulnerability

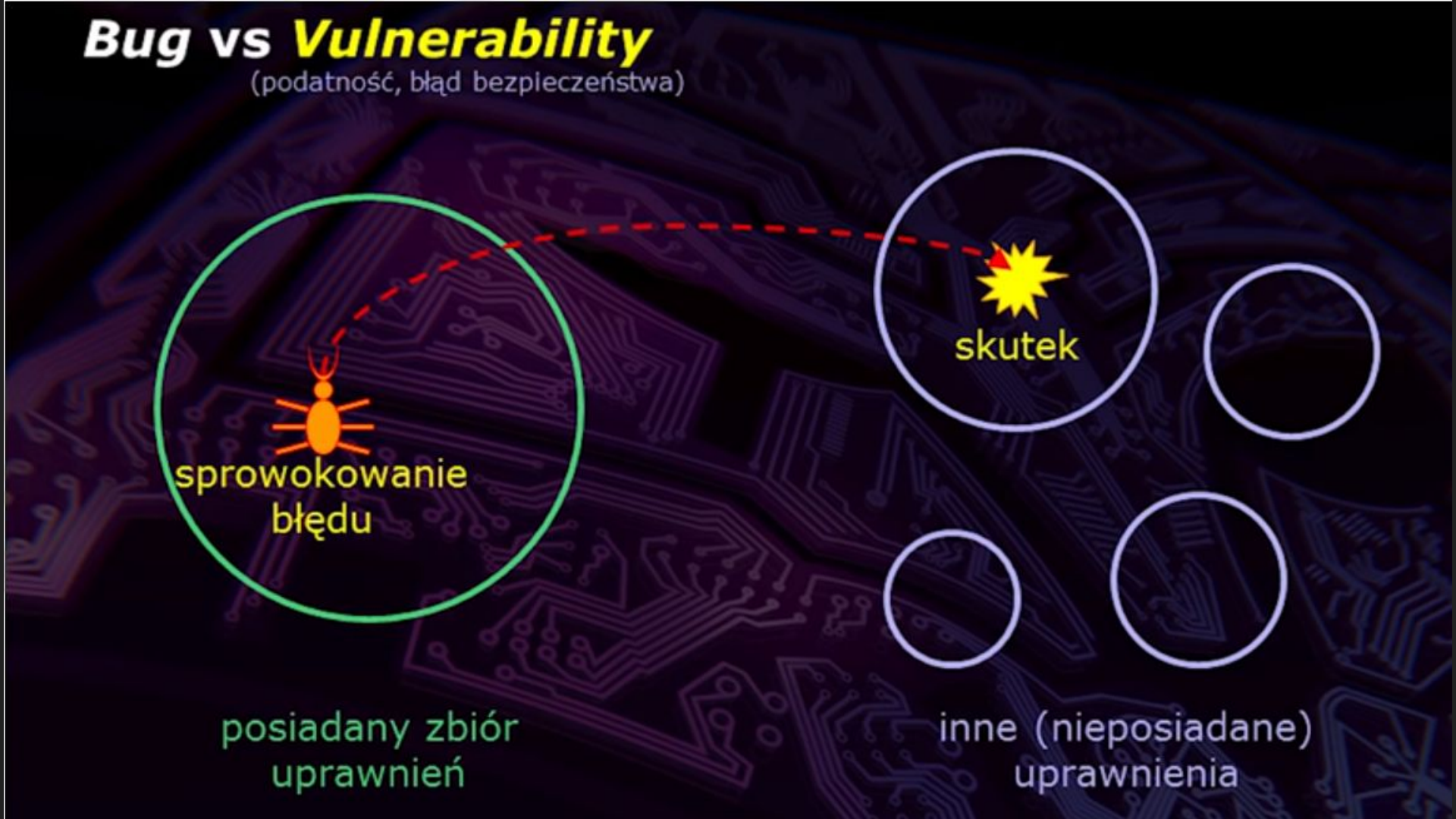
(podatność, błąd bezpieczeństwa)

posiadany zbiór
uprawnień

inne (nieposiadane)
uprawnienia

Bug vs Vulnerability

(podatność, błąd bezpieczeństwa)



Common attack techniques

- ROP
- buffer overflow
- format string attack
- shellcode injection
- SROP
- ...

Mitigations

- NX stack aka W^X
- PIE
- (partial) RELRO
- CANARY
- separate local stack variables
- ...

Common attack techniques

- ROP
- buffer overflow
- format string attack
- shellcode injection
- SROP
- ...

Mitigations

- NX stack aka W^X
- PIE
- (partial) RELRO
- CANARY
- separate local stack variables
- ...

Common attack techniques

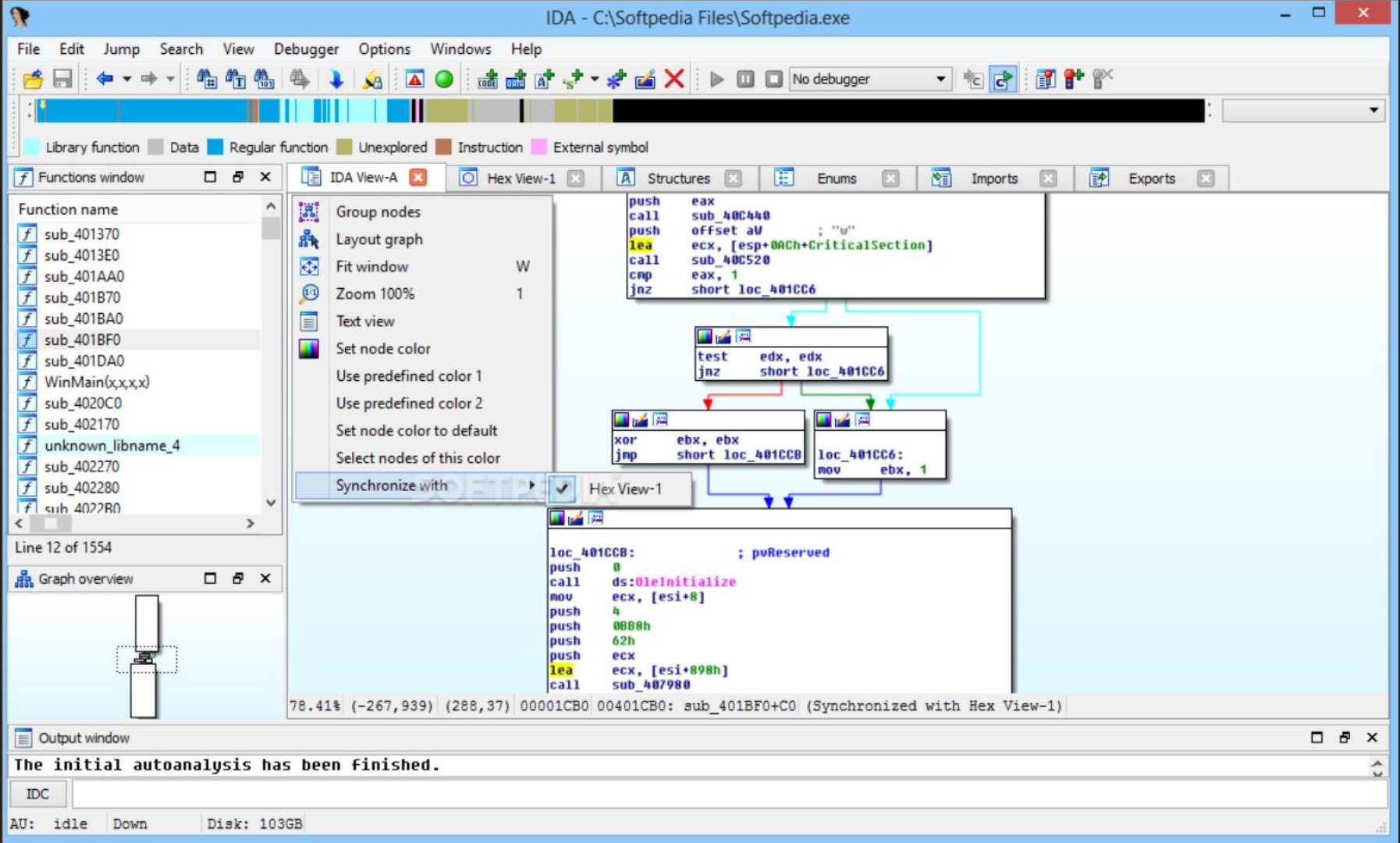
- ROP
- buffer overflow
- format string attack
- shellcode injection
- SROP
- ...

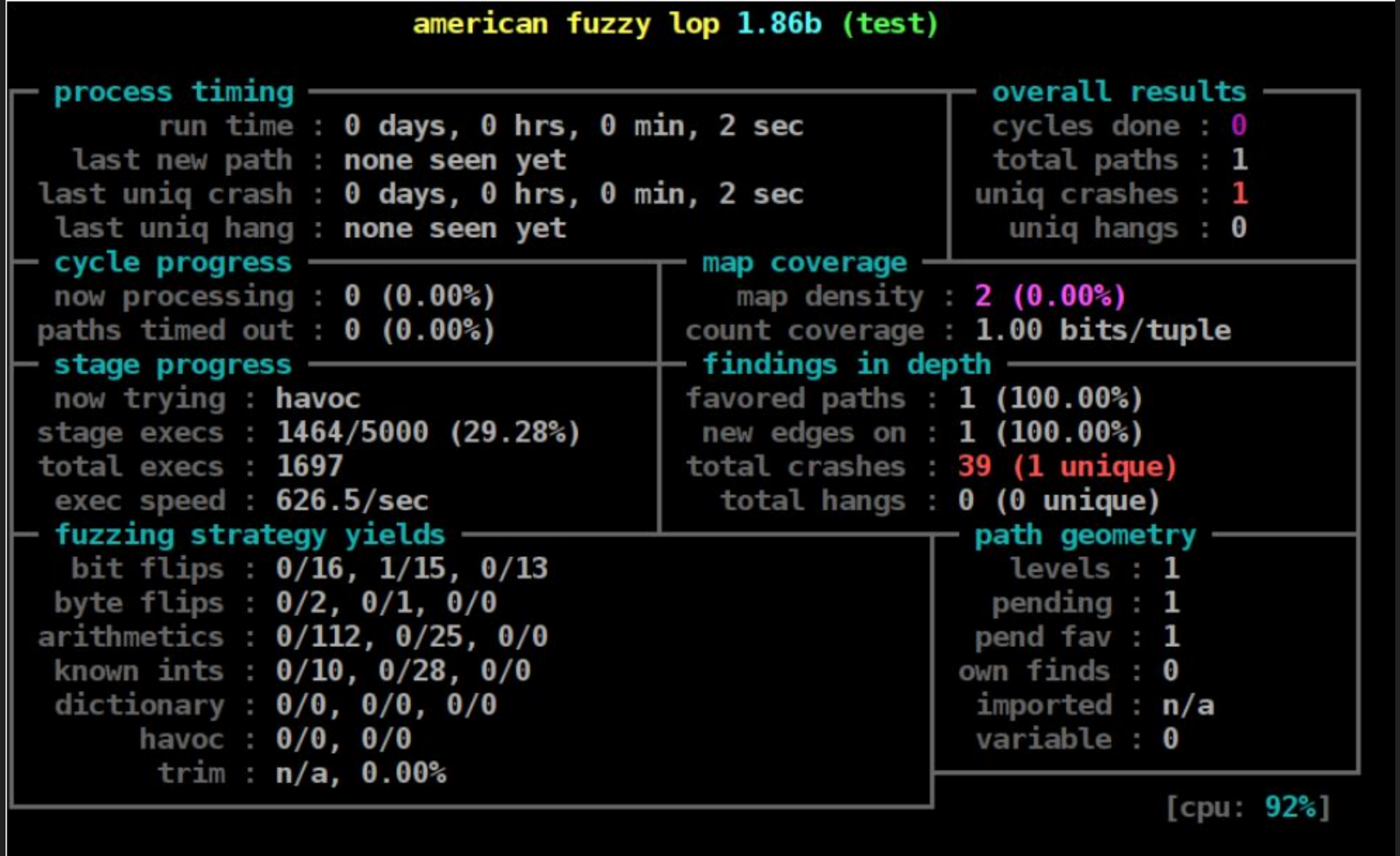
Mitigations

- NX stack aka W^X
- PIE
- (partial) RELRO
- CANARY
- separate local stack variables
- ...

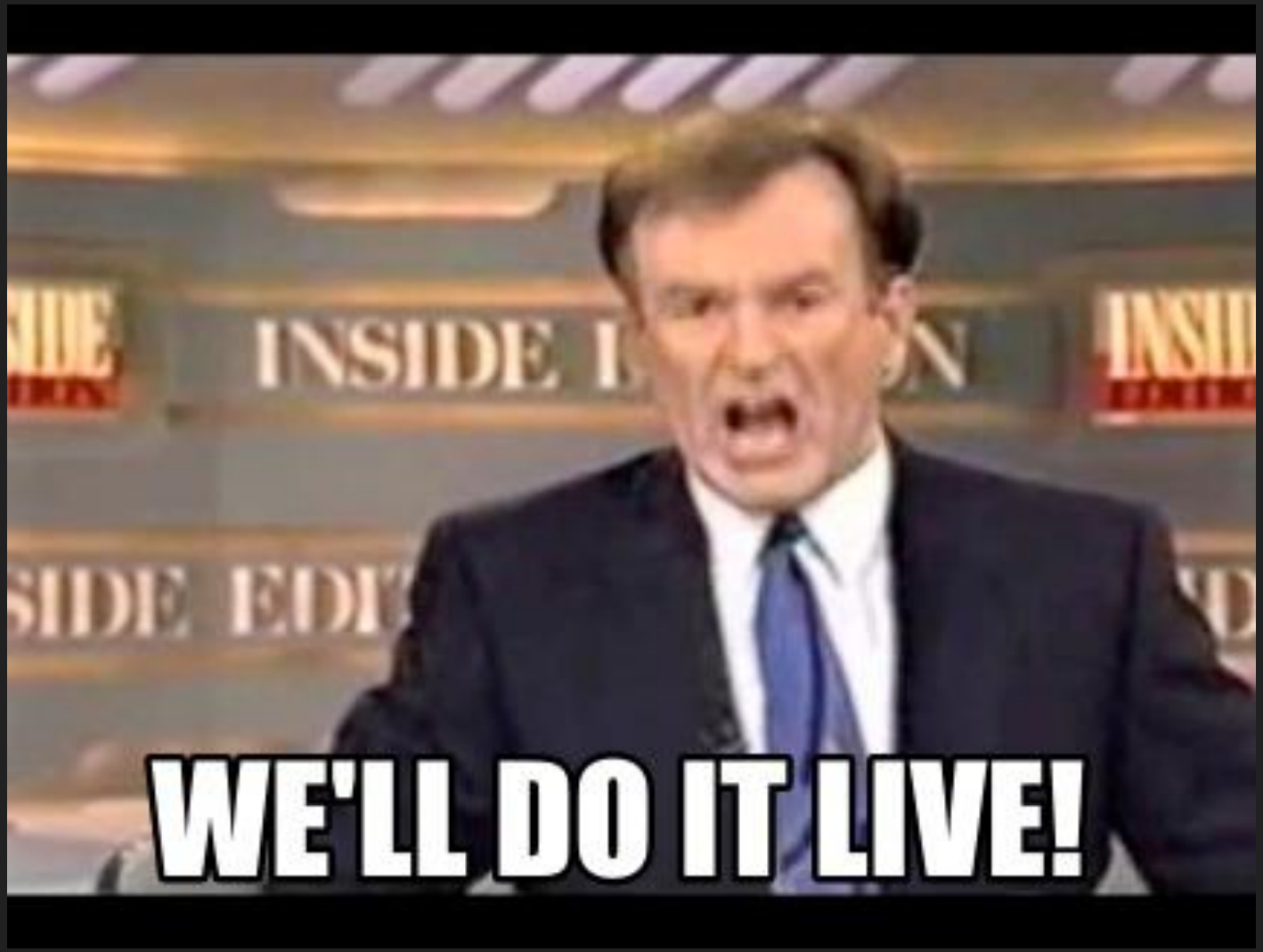
Tools

- Ida-Pro
- Ghidra
- radare2
- strace
- ltrace
- gdb -> pwndbg, gef, peda
- pwnlib & pwntools
- one-gadget
- ropper & ROPgadget
- american fuzzy lop
- angr & angrop
- metasploit
- ...
- many
- MANY
- MORE


IDA - C:\Softpedia Files\Softpedia.exe
File Edit Jump Search View Debugger Options Windows Help
No debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name
sub_401370
sub_4013E0
sub_401AA0
sub_401B70
sub_401BA0
sub_401BF0
sub_401DA0
WinMain(x,x,x,x)
sub_4020C0
sub_402170
unknown_libname_4
sub_402270
sub_402280
Line 12 of 1554
Graph overview
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports
Group nodes
Layout graph
Fit window W
Zoom 100% 1
Text view
Set node color
Use predefined color 1
Use predefined color 2
Set node color to default
Select nodes of this color
Synchronize with
Hex View-1
push eax
call sub_40C440
push offset aW ; "W"
lea ecx, [esp+0ACh+CriticalSection]
call sub_40C520
cmp eax, 1
jnz short loc_401CC6
test edx, edx
jnz short loc_401CC6
xor ebx, ebx
jmp short loc_401CCB
loc_401CC6:
mov ebx, 1
loc_401CCB: ; pvReserved
push 0
call ds:OleInitialize
mov ecx, [esi+8]
push 4
push 0BB8h
push 62h
push ecx
lea ecx, [esi+898h]
call sub_407980
78.41% (-267,939) (288,37) 00001CB0 00401CB0: sub_401BF0+C0 (Synchronized with Hex View-1)
Output window
The initial autoanalysis has been finished.
IDC
AU: idle Down Disk: 103GB


american fuzzy lop 1.86b (test)
process timing
run time : 0 days, 0 hrs, 0 min, 2 sec
last new path : none seen yet
last uniq crash : 0 days, 0 hrs, 0 min, 2 sec
last uniq hang : none seen yet
overall results
cycles done : 0
total paths : 1
uniq crashes : 1
uniq hangs : 0
cycle progress
now processing : 0 (0.00%)
paths timed out : 0 (0.00%)
map coverage
map density : 2 (0.00%)
count coverage : 1.00 bits/tuple
stage progress
now trying : havoc
stage execs : 1464/5000 (29.28%)
total execs : 1697
exec speed : 626.5/sec
findings in depth
favored paths : 1 (100.00%)
new edges on : 1 (100.00%)
total crashes : 39 (1 unique)
total hangs : 0 (0 unique)
fuzzing strategy yields
bit flips : 0/16, 1/15, 0/13
byte flips : 0/2, 0/1, 0/0
arithmetics : 0/112, 0/25, 0/0
known ints : 0/10, 0/28, 0/0
dictionary : 0/0, 0/0, 0/0
havoc : 0/0, 0/0
trim : n/a, 0.00%
path geometry
levels : 1
pending : 1
pend fav : 1
own finds : 0
imported : n/a
variable : 0
[cpu: 92%]

```
*RAX  0x1c
 RBX  0x0
*RCX  0x7fffffffeca8 —▸ 0x7fffffffeeaf ◂— 0x4f494e4f48545950 ('PYTHONIO')
*RDX  0x7ffff7de8a50 (_dl_fini) ◂— push   rbp
*RDI  0x7ffff7ffe168 ◂— 0x0
*RSI  0x1
*R8   0x7ffff7ffe6f8 ◂— 0x0
 R9   0x0
 R10  0x0
*R11  0x1
*R12  0x4006b0 ◂— xor    ebp, ebp
*R13  0x7fffffffec90 ◂— 0x1
 R14  0x0
 R15  0x0
 RBP  0x0
 RSP  0x7fffffffec90 ◂— 0x1
*RIP  0x4006b0 ◂— xor    ebp, ebp
[──────────────────────────────────────────────────DISASM──────────────────────────────────────────────────]
 ► 0x4006b0    xor    ebp, ebp
   0x4006b2    mov    r9, rdx
   0x4006b5    pop    rsi
   0x4006b6    mov    rdx, rsp
   0x4006b9    and    rsp, 0xfffffffffffffff0
   0x4006bd    push   rax
   0x4006be    push   rsp
   0x4006bf    mov    r8, 0x4009c0
   0x4006c6    mov    rcx, 0x400950
   0x4006cd    mov    rdi, 0x4007a6
   0x4006d4    call   0x400680
[──────────────────────────────────────────────────STACK───────────────────────────────────────────────────]
00:0000│ r13 rsp  0x7fffffffec90 ◂— 0x1
01:0008│          0x7fffffffec98 —▸ 0x7fffffffee7b ◂— 0x2f6465726168532f ('/Shared/')
02:0010│          0x7fffffffeca0 ◂— 0x0
03:0018│ rcx      0x7fffffffeca8 —▸ 0x7fffffffeeaf ◂— 0x4f494e4f48545950 ('PYTHONIO')
04:0020│          0x7fffffffecb0 —▸ 0x7fffffffeec6 ◂— 0x79786f72705f6f6e ('no_proxy')
05:0028│          0x7fffffffecb8 —▸ 0x7fffffffeee3 ◂— 0x454d414e54534f48 ('HOSTNAME')
06:0030│          0x7fffffffecc0 —▸ 0x7fffffffeef9 ◂— 0x313d4c564c4853 /* 'SHLVL=1' */
07:0038│          0x7fffffffecc8 —▸ 0x7fffffffef01 ◂— 'HOME=/root'
[─────────────────────────────────────────────────BACKTRACE──────────────────────────────────────────────────]
 ► f 0           4006b0
   f 1              1
   f 2     7fffffffee7b
   f 3              0
Breakpoint *0x4006b0
pwndbg>
```


WE'LL DO IT LIVE!

- https://trailofbits.github.io