Strategic Surprise
My name is Nico Waisman and I have an obsession
loc_7C931B39:
    imul    eax, 2C98h
    lea    eax, [eax+ebx+90Dh] ; Heap Local Data
    mov    [ebp+LocalData], eax
    imul    esi, 58h ; BucketSize=3x58 (Segment size)
    lea    eax, [esi+eax+88h]
    mov    [ebp+SegmentInfo], eax
    mov    ebx, [eax+_HEAP_LOCAL_SEGMENT_INFO.Hint]
    mov    [ebp+SubSegment], ebx
    test    ebx, ebx
    jz    RETRY_ALLOC

loc_7C931D14:
    mov    ecx, ebx
    call    @ExInterlockedPopEntrySList@@8 ; ExInterlockedPopEntrySList(x,x)
    test    eax, eax
    jnz    loc_7C94F088

loc_7C94F088:
    and    [ebp+ms_exc.disabled], 0
    mov    esi, [ebp+HeapBucket]

loc_7C94F0B0:
    xor    ebx, ebx
    lea    esi, [eax-18h]
    lea    ecx, [esi+1Ch]

loc_7C931B6A:
    ; AggregateExchg
    mov    eax, [ebx+8]
    mov    [ebp+var_76], eax
    mov    BlockSize, [ebx+7Ch] ; Block Size
    mov    [ebp+var_78], BlockSize
    mov    [ebp+var_80], eax
    test    ax, ax ; if SubSegment->Depth > 0
    jz    loc_7C93224F
Once upon a time in Mexico...
zb0@podridito:~/research/bug-lbfd# cat test.c
int main()
{
    printf("IRA MA\n");
}
zb0@podridito:~/research/bug-lbfd# gcc test.c -o test
zb0@podridito:~/research/bug-lbfd# ./mtraveca0 -f test
looking for section table at: 7e0
size: 8193 - alargando a: 12289
284-
804a5b4 804a498
shellcode @ 0x080d7828 (281)
zb0@podridito:~/research/bug-lbfd# /usr/local/bin/objdump -R test
/usr/local/bin/objdump: test: Invalid argument
Did you teletubbie?

test: file format elf32-i386

DYNAMIC RELOCATION RECORDS
OFFSET TYPE VALUE
08049574 R_386_GLOB_DAT __gmon_start__
08049564 R_386_JUMP_SLOT __register_frame_info
08049568 R_386_JUMP_SLOT __deregister_frame_info
0804956c R_386_JUMP_SLOT __libc_start_main
08049570 R_386_JUMP_SLOT printf

zb0@podridito:~/research/bug-lbfd# /usr/local/bin/objdump -R test

test: file format elf32-i386

DYNAMIC RELOCATION RECORDS
OFFSET TYPE VALUE
08049574 R_386_GLOB_DAT __gmon_start__
08049564 R_386_JUMP_SLOT __register_frame_info
08049568 R_386_JUMP_SLOT __deregister_frame_info
0804956c R_386_JUMP_SLOT __libc_start_main
08049570 R_386_JUMP_SLOT printf

zb0@podridito:~/research/bug-lbfd#
Advance Doug Lea’s malloc exploits

Vudo - An object superstitiously believed to embody magical powers

Once upon a free()
Wuftp g106/site exec
G0bbles openssh exploit (FUCKYOUTHEO)
Pserverd - 4c1d6tch3z
Solar Designer Netscape JPEG exploit
Understand your exploitation domain
Reverse, Reverse and Reverse a little bit more
Debug, Debug and Debug a little bit more
Five Ws
The Present
I've been looking into starting to bindiff Patch Tuesday patches in order to get used to finding sploits just haven't had the time. Any recommendations on windows differing tools?

save yourself the heartache and frustration, just kill yourself now

haha :P

on *nix systems I'm good with diff/patch I'm just ignorant to the windows equiv.

its nowhere near the same game

that drastic of a change eh?

you spend an hour just getting the patches extracted, another half and hour loading into ida, then umpteen thousand hours going through tens of thousands of changed functions based on the asm alone

and when you finally win, you only know what changed, not even how to trigger it

so you spend another umpteen hours trying to trace the code path to that changed bit of code

only to realize 2 days later it was the wrong change and the one you're looking at is unexploitable

then you kill yourself

so like jduck said, skip a step :P
Why is HDMoore Sad?

Disclaimer: The imagery used in this Slide may have been altered or modified to some degree from the original image.
Exploits are hard...
When was the last time you saw a real public exploit?
Post Mortem Exploits
Dry Humping Exploits
@0xcharlie
Charlie Miller

Reduced from high precision heap manipulation to just "trying stuff" in iPhone exploit. #somebugssuck
Hope is not a
Business Plan
Exploits are hard...

...but it was always being
The Element of Surprise
Team vs Individuals
Researchers
You don’t need a researcher, you need a unicorn.
Programmer
LAB
Management
Windows 2000

1d: Triggering the bug
2-4d: Understanding the heap layout
2-5d: Finding Soft and Hard Memleaks
5-8d : Finding a reliable Write4
1-2d: Function Pointers and Shellcode

Windows Vista

1 d: Triggering the bug
1-2d: Understanding the heap layout
2-5d: Finding Soft and Hard Memleaks
10-30d : Overwriting a the correct memory
2-5 days: Function pointer and Shellcode
Windows 2000

1d: Triggering the bug

2-4d: Understanding the heap layout

2-5d: Finding Soft and Hard Memleaks

5-8d: Finding a reliable Write4

1-2d: Function Pointers and Shellcode

Windows Vista

1d: Triggering the bug

1-2d: Understanding the heap layout

2-5d: Finding Soft and Hard Memleaks

10-30d: Overwriting a the correct memory

2-5 days: Function pointer and Shellcode
Protection never target what we always aim for...
Bug classes die
Primitives dont
Nowadays exploitation techniques are crumbs of the '90 great banquet
There are No Surprises
Questions?

nico@immunityinc.com

@nicowaismann