

(S\$\$)

ixia



Hexcellents

Session 4

Binary Executable Analysis

Security Summer School
3rd of July 2014
ACS/Ixia/Hexcellents

Binary Executables

- Non-text files
 - contain non-human-readable characters
- Executed by computers
- Formats
 - Linux - ELF (Executable Linking Format)
 - Windows - PE (Portable Executable)

Generating Binary Files

Source Code

```
int main()
{
    return 0;
}
```

compile

Assembly Code

```
main:
    push    ebp
    mov     ebp, esp
    mov     eax, 0
    pop     ebp
    ret
```

assemble

Machine Code

```
0000000: 55 89 e5 b8
0000004: 00 00 00 00
0000008: 5d c3
```

execute!

Restoring Source Files

Machine Code

```
0000000: 55 89 e5 b8  
0000004: 00 00 00 00  
0000008: 5d c3
```


disassemble



Assembly Code

```
main:  
    push    ebp  
    mov     ebp, esp  
    mov     eax, 0  
    pop     ebp  
    ret
```

decompile



Source Code

```
int main()  
{  
    return 0;  
}
```

read?

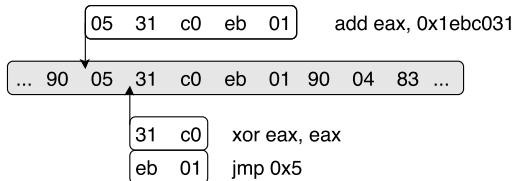
Disassembly

- Retrieve the assembly code from the machine code
- Static analysis
- Linear sweep - objdump
- Recursive traversal - IDA

Disassembly - Limitations

- Instructions
 - have variable lengths
 - can overlap
 - can be unaligned

Example



Linear Sweep

- Starts from the beginning of each section
- Decode one instruction after another
- + parses the entire program
- - cannot detect unaligned instructions

Recursive Traversal

- Starts from the program's main entry point
- Performs linear sweep until a branch/jump instruction is met
- Follows the control instructions in a recursive manner
- Similar to control flow analysis
- + parses only code that is explicitly hit
- - cannot handle indirect branches, PIC

What next?

- Disassembly is not 100% precise
- Static analysis cannot detect all execution paths
- For everything else -

What next?

- Disassembly is not 100% precise
- Static analysis cannot detect all execution paths
- For everything else - there's GDB

GDB

```
Dump of assembler code for function main:
0x08048596 <+0>:  push  ebp
0x08048597 <+1>:  mov   ebp,esp
0x08048599 <+3>:  sub   esp,0x18
0x0804859c <+6>:  and   esp,0xffffffff
0x0804859f <+9>:  mov   eax,0x0
0x080485a4 <+14>: sub   esp,eax
0x080485a6 <+16>: cmp   DWORD PTR [ebp+0x8],0x2
0x080485aa <+20>: je    0x80485ca <main+52>
0x080485ac <+22>: mov   eax,DWORD PTR [ebp+0xc]
0x080485af <+25>: mov   eax,DWORD PTR [eax]
0x080485b1 <+27>: mov   DWORD PTR [esp+0x4],eax
0x080485b5 <+31>: mov   DWORD PTR [esp],0x8048760
0x080485bc <+38>: call  0x80483b8 <printf@plt>
0x080485c1 <+43>: mov   DWORD PTR [ebp-0x4],0x0
0x080485c8 <+50>: jmp   0x8048618 <main+130>
0x080485ca <+52>: call  0x804852d <pass>
0x080485cf <+57>: mov   DWORD PTR [esp+0x8],0x64
0x080485d7 <+65>: mov   eax,DWORD PTR [ebp+0xc]
0x080485da <+68>: add   eax,0x4
0x080485dd <+71>: mov   eax,DWORD PTR [eax]
0x080485df <+73>: mov   DWORD PTR [esp+0x4],eax
0x080485e3 <+77>: mov   DWORD PTR [esp],0x80491a0
0x080485ea <+84>: call  0x80483a8 <mbstowcs@plt>
0x080485ef <+89>: mov   DWORD PTR [esp+0x4],0x8049140
0x080485f7 <+97>: mov   DWORD PTR [esp],0x80491a0
0x080485fe <+104>: call  0x80483d8 <wcscmp@plt>
0x08048603 <+109>: test  eax,eax
0x08048605 <+111>: jne   0x804860c <main+118>
0x08048607 <+113>: call  0x80484b4 <win>
0x0804860c <+118>: mov   DWORD PTR [esp],0x8048795
0x08048613 <+125>: call  0x80483e8 <puts@plt>
0x08048618 <+130>: mov   eax,DWORD PTR [ebp-0x4]
0x0804861b <+133>: leave
0x0804861c <+134>: ret

End of assembler dump.
```

GDB++

```
Dump of assembler code for function main:
0x08048596 <+0>:  push  ebp
0x08048597 <+1>:  mov   ebp,esp
0x08048599 <+3>:  sub   esp,0x18
0x0804859c <+6>:  and   esp,0xfffffff0
0x0804859f <+9>:  mov   eax,0x0
0x080485a4 <+14>: sub   esp,eax
0x080485a6 <+16>:  cmp   DWORD PTR [ebp+0x8],0x2
0x080485aa <+20>:  je    0x80485ca <main+52>
0x080485ac <+22>:  mov   eax,DWORD PTR [ebp+0xc]
0x080485af <+25>:  mov   eax,DWORD PTR [eax]
0x080485b1 <+27>:  mov   DWORD PTR [esp+0x4],eax
0x080485b5 <+31>:  mov   DWORD PTR [esp],0x8048760
0x080485bc <+38>:  call  0x80483b8 <printf@plt>
0x080485c1 <+43>:  mov   DWORD PTR [ebp-0x4],0x0
0x080485c8 <+50>:  jmp   0x8048618 <main+130>
0x080485ca <+52>:  call  0x804852d <pass>
0x080485cf <+57>:  mov   DWORD PTR [esp+0x8],0x64
0x080485d7 <+65>:  mov   eax,DWORD PTR [ebp+0xc]
0x080485da <+68>:  add   eax,0x4
0x080485dd <+71>:  mov   eax,DWORD PTR [eax]
0x080485df <+73>:  mov   DWORD PTR [esp+0x4],eax
0x080485e3 <+77>:  mov   DWORD PTR [esp],0x80491a0
0x080485ea <+84>:  call  0x80483a8 <mbstowcs@plt>
0x080485ef <+89>:  mov   DWORD PTR [esp+0x4],0x8049140
0x080485f7 <+97>:  mov   DWORD PTR [esp],0x80491a0
0x080485fe <+104>: call  0x80483d8 <wcscmp@plt>
0x08048603 <+109>: test  eax,eax
0x08048605 <+111>: jne   0x804860c <main+118>
0x08048607 <+113>: call  0x80484b4 <win>
0x0804860c <+118>: mov   DWORD PTR [esp],0x8048795
0x08048613 <+125>: call  0x80483e8 <puts@plt>
0x08048618 <+130>: mov   eax,DWORD PTR [ebp-0x4]
0x0804861b <+133>: leave
0x0804861c <+134>: ret
```

End of assembler dump.

Runtime analysis

```
root@dmns:x86_64 [tmp] # gdb -q ./service_xyz
Reading symbols from ./service_xyz...(no debugging symbols found)...done.
(gdb) b *main
Breakpoint 1 at 0x8048ee9
(gdb) run
Starting program: /tmp/tmp/service_xyz
warning: the debug information found in "/usr/lib64/debug/lib64/ld-2.17.so."
is not accurate.
warning: Could not load shared library symbols for linux-gate.so.1.
Do you need "set solib-search-path" or "set sysroot"?

Breakpoint 1, 0x08048ee9 in main ()
(gdb) info reg
eax                0x1          1
ecx                0xffffce84   -12668
edx                0x8048ee9    134516457
ebx                0xf7f94e54   -134656428
esp                0xffffcdec   0xffffcdec
ebp                0x0          0x0
esi                0x0          0
edi                0x0          0
eip                0x8048ee9    0x8048ee9 <main>
eflags            0x246        [ PF ZF IF ]
cs                 0x23         35
ss                 0x2b         43
ds                 0x2b         43
es                 0x2b         43
fs                 0x0          0
gs                 0x63         99
(gdb) █
```

Runtime analysis++

```
[-----registers-----]
EAX: 0x1
EBX: 0xf7f98e54 --> 0x1a6d5c
ECX: 0xffffcfe4 --> 0xffffd1ae ("/tmp/smash/level01")
EDX: 0x8048596 (<main>: push ebp)
ESI: 0x0
EDI: 0x0
EBP: 0x0
ESP: 0xffffcf4c --> 0xf7e0ece5 (<__libc_start_main+245>: mov DWORD PTR [esp],eax)
EIP: 0x8048596 (<main>: push ebp)
EFLAGS: 0x246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[-----code-----]
0x804858a <pass+93>: mov DWORD PTR ds:0x8049160,0x0
0x8048594 <pass+103>: leave
0x8048595 <pass+104>: ret
=> 0x8048596 <main>: push ebp
0x8048597 <main+1>: mov ebp,esp
0x8048599 <main+3>: sub esp,0x18
0x804859c <main+6>: and esp,0xffffffff
0x804859f <main+9>: mov eax,0x0
[-----stack-----]
0000| 0xffffcf4c --> 0xf7e0ece5 (<__libc_start_main+245>: mov DWORD PTR [esp],eax)
0004| 0xffffcf50 --> 0x1
0008| 0xffffcf54 --> 0xffffcfe4 --> 0xffffd1ae ("/tmp/smash/level01")
0012| 0xffffcf58 --> 0xffffcfec --> 0xffffd1c1 ("MANPATH=/etc/java-config-2/current-system-
l/share/man:/usr/share/man:/usr/share/gcc-data/x86_64-pc-linux-gnu/4.7.3/man:/usr/share/bin
4-pc-linux-gnu/2.23.2/man:/etc/java"...)
0016| 0xffffcf5c --> 0xf7ffcfc0 --> 0x20ef8
0020| 0xffffcf60 --> 0x1c
0024| 0xffffcf64 --> 0x0
0028| 0xffffcf68 --> 0xf7fda4c8 --> 0xf7df2000 --> 0x464c457f
[-----]
Legend: code, data, rodata, value
Breakpoint 1, 0x08048596 in main ()
```

Keywords

- executable
- PE
- ELF
- compile
- assemble
- decompile
- disassembly
- IDA
- objdump
- gdb

Resources

- <https://www.hex-rays.com/products/ida/>