

Cyber@UC Meeting 85

Battelle goat challenge/IDA

If You're New!

- Join our Slack: cyberatuc.slack.com
- Check out our website: cyberatuc.org
- Organization Resources on our Wiki: wiki.cyberatuc.org
- **SIGN IN!** (*Slackbot will post the link in #general every Wed@6:30*)
- Feel free to get involved with one of our committees:
Content Finance Public Affairs Outreach Recruitment Lab
- Ongoing work in our research lab!



Announcements

- Looking for **lab committee** volunteers!
- New bi-weekly lab events!
- Grilled Cheese at Baldwin was lit!
- Executive meeting sunday, all are welcome to come
- Revolution UC coming up!
- Dodgeball!
- Upcoming Loveland outreach march 11th
- Smash after meeting!!!!!!



Workshop: Goat Disassembly

The Topics Today Go Something Exactly Like This

- Quick touch on Assembly & Disassembly
- The RE tools in Kali and IDA
- Battelle's *Feed the Magical Goat* CTF




Assembly?!

- Nearest possible human readable version of machine code
- Everything is either stored in registers, which can be compared to variables, or in literals values (ints/strings)
- Functions are called subprocesses
- First years take note

08048918	pushl	%ebp
08048919	movl	%esp, %ebp
0804891b	subl	\$0x4, %esp
0804891e	movl	\$0x0, 0xfffffffffc(%ebp)
08048925	cmpl	\$0x63, 0xfffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xfffffffffc(%ebp), %eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8, %esp
08048941	incl	0xfffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax, %eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	

Registers?!

- Usually prefixed with a “%”
- You only have 8 that you should really be looking at / using
- Basically 32 bit pointers / ints
 - Pointers are ints
- Google the names for x64, there's plenty of tables



08048918	pushl	%ebp
08048919	movl	%esp, %ebp
0804891b	subl	\$0x4, %esp
0804891e	movl	\$0x0, 0xfffffffffc(%ebp)
08048925	cmpl	\$0x63, 0xfffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xfffffffffc(%ebp), %eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8, %esp
08048941	incl	0xfffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax, %eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	

Subprocesses

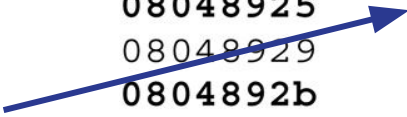
- Equivalent of functions
- Functions arguments are **pushed** onto the stack
- The subprocess is **called**
- Subprocess **return** as functions do

08048918	pushl	%ebp
08048919	movl	%esp,%ebp
0804891b	subl	\$0x4,%esp
0804891e	movl	\$0x0,0xffffffffc(%ebp)
08048925	cmpl	\$0x63,0xffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xffffffffc(%ebp),%eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8,%esp
08048941	incl	0xffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax,%eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	


Conditionals

- Variables can be **compared**
- **Jumps** in execution can be made depending on comparisons
- Jumps can also be unconditional (like goto & break)
- C if statements are typically compares and jumps sequentially executed

08048918	pushl	%ebp
08048919	movl	%esp,%ebp
0804891b	subl	\$0x4,%esp
0804891e	movl	\$0x0,0xffffffffc(%ebp)
08048925	cmpl	\$0x63,0xffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xffffffffc(%ebp),%eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8,%esp
08048941	incl	0xffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax,%eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	



Other Notes

- Strings are typically stored as static character arrays then copied later when they are used
- This is basically just C with harder syntax and heavy use of goto
- Every instruction has a position offset value compared to where the program's base memory address is 

08048918	pushl	%ebp
08048919	movl	%esp, %ebp
0804891b	subl	\$0x4, %esp
0804891e	movl	\$0x0, 0xfffffffffc(%ebp)
08048925	cmpl	\$0x63, 0xfffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xfffffffffc(%ebp), %eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8, %esp
08048941	incl	0xfffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax, %eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	

Other Notes Cont.

- AT&T vs Intel Format
- **Move** operations just copy paste a register value into another register

08048918	pushl	%ebp
08048919	movl	%esp,%ebp
0804891b	subl	\$0x4,%esp
0804891e	movl	\$0x0,0xfffffffffc(%ebp)
08048925	cmpl	\$0x63,0xfffffffffc(%ebp)
08048929	jle	08048930
0804892b	jmp	08048948
0804892d	nop	
0804892e	nop	
0804892f	nop	
08048930	movl	0xfffffffffc(%ebp),%eax
08048933	pushl	%eax
08048934	pushl	\$0x8049418
08048939	call	080487c0 <printf>
0804893e	addl	\$0x8,%esp
08048941	incl	0xfffffffffc(%ebp)
08048944	jmp	08048925
08048946	nop	
08048947	nop	
08048948	xorl	%eax,%eax
0804894a	jmp	0804894c
0804894c	leave	
0804894d	ret	

Disassembly

- All the 1337 HaX0rs do it
- You should too
- Process of taking apart binary programs, which are typically compiled from C/C++
- **Static analysis** - Just reading assembly code
- **Dynamic analysis** - running and debugging the program
- Basically just feed a binary in and assembly code comes out



ComputerHope.com

Disassembly Tools in Kali Linux (and IDA)


Binary Tools (ELF / PE)	Android / Java Tools
diStorm3 IDA edb-debugger OllyDbg Valgrind YARA strings	apktool dex2jar jad javasnoop jd-gui smali



Interactive Disassembler (IDA)

- Download the **free** version from <https://www.hex-rays.com/>
- Grab the Magical Goat zip file from <https://www.battelle.org/cyber-challenge>
- I don't have any slides for IDA itself so we'll just go into it with the binary

Alternatives to IDA:

- Radare2 (r2)
 - Binary Ninja, which has really nice intermediate language support
 - GHIDRA, the NSA made equivalent to be released in March
- 



BATTELLE