				edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84] esi esi eax edi [ebp+arg_0], eax sub_31486A eax, eax	
	Syllabus and	Revi	jz push lea push wush push call test jz cmp	<pre>short loc_31306D esi eax, [ebp+arg_0] eax i, 1D0h [ebp+arg_4] edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi</pre>	
	Modern Binary Exp	oloitat		short loc_31308F	
	CSCI 4968 - Sprin			0Dh sub_31411B	
	Alex Bulaze	el			
		loc_31307D:	call and	sub_3140F3 eax, OFFFFh	CODE XREF: sub_312FD8
MBE - 01/27/2015	Syllabus and Review	loc_31308C:			1 CODE XREF: sub_312FD8

Lecture Overview

1. Syllabus

- 2. Course Overview
- 3. Review of Background Material
 - a. Linux
 - b. C
 - c. x86 Assembly

		sub_314623
		short loc_31306D
		[ebp+arg_0], ebx
		short loc_313066
		eax, [ebp+var_70]
		eax, [ebp+var_84]
		short loc_313066
		eax, [ebp+var 84]
	pusn	0651
		[ebp+arg_0], eax
		sub_31486A
		short loc_31306D
		eax, [ebp+arg_0]
al		
ai		[ebp+arg_4]
		sub_314623
		short loc 31306D
		[ebp+arg_0], esi
		short loc_31308F
3066:		
		sub_31411B
		sub_3140F3
		short loc_31307D
		sub_3140F3
		short loc_31308C
307D:		
	call	sub_3140F3
		eax, OFFFFh
		eax, 80070000h
		2

Syllabus and Review

Course Details

	•	Modern	Binary	Exp	loitation
--	---	--------	--------	-----	-----------

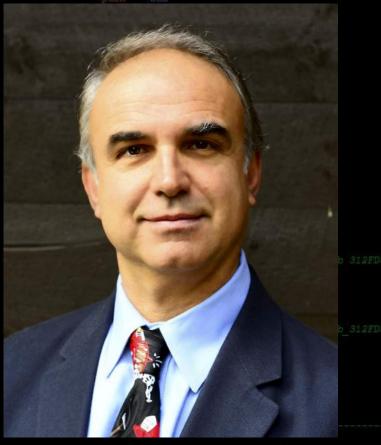
- Course Number: CSCI 4968
- Credit Hours: 4
- Semester / Year: Spring 2015
- Meeting Days: Tuesday/Friday 2-4PM
- Room Location: Walker 5113
- Course Website:
 - http://security.cs.rpi.edu/courses/binexp-spring2015/
 - http://rpis.ec/binexp
- Prereqs:
 - CSCI 2500 Computer Organization
 - ECSE 2660 Computer Architecture, Networks, and Operating Systems

		loc_31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				3
		loc_31308C:			
				[ebp+var 4], ea	

Instructor

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
pusn	051

- Instructor: Dr. Bülent Yener
 - Office: Lally 310
 - Email: yener@cs.rpi.edu



			n		
U			ч		

Syllabus and Review

loc_31308C:

; CODE XREF: sub_312 +var 41. eax

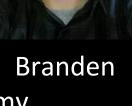
Cyber Is A Team Sport

sub_314623
short loc_31306D
[ebp+arg_0], ebx
short loc_313066
<pre>eax, [ebp+var_70]</pre>
<pre>eax, [ebp+var_84]</pre>
short loc_313066
<pre>eax, [ebp+var_84]</pre>
881





Markus Jeremy





Sophia

Austin



Alex

5

Syllabus and Review



MBE - 01/27/2015







Office Hours

- Office hours:
 - Wednesday 7-10 PM @ Sage 3101
- Come hang out at RPISEC hack nights!
 - Ask questions, get extra help with MBE
 - Collaborate on HW/Labs
 - Work on security projects, challenges, etc

				short loc_313080	
		loc_31307D:		sub 3140F3	
IBE - 01/27/2015	Syllabus and Review		and or	eax, OFFFFh eax, 80070000h	6
	Synabas and Review	loc_31308C:			; CODE XREF: sub_312FD8

Othe	r Options				di ub_314623 ax, eax hort loc_31306D ebp+arg_0], ebx hort loc_313066 ax, [ebp+var_70] ax, [ebp+var_84] hort loc_313066 ax, [ebp+var_84] si
SORRY FOR THE VOICEMAIL, BUT I'M CONFUSED ABOUT HOW TO REACH YOU.	WHEN I TEXT YOU, YOU REPLY ONCE ON GCHAT, THEN GO QUIET, YET ANSWER IRC RIGHT AWAY. I EMAILED YOU, AND YOU REPLIED ON SKYPE AND MENTIONED THAT THE EMAIL "WOKE YOU UP."	YOU'RE VERY RESPONSIVE- I JUST HAVE NO SENSE OF HOW YOU USE TECHNOLOGY.	5!? 5!?		and and and and and and and and
/ \	/ \	$ 1 \rangle$	/ \ ; loc_31307D:		ub_3140F3 hort loc_31308C
MBE - 01/27/2015	Syllal	bus and Review	loc_31307D:	and e	, CODE XREF: SUB 312FD2 ax, OFFFFh ax, 80070000h ; CODE XREF: Sub 312FD2 ; CODE XREF: Sub 312FD2

Digital Office Hours (IRC)

- The RPISEC IRC http://rpis.ec/irc
 - server: irc.rpis.ec
 - port: 6667 (6697 for SSL)
 - room: #rpisec

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	eax, [ebp+var_70]
	eax, [ebp+var 84]
	short loc_313066
	eax, [ebp+var_84]
pusn	esi
	[ebp+arg_0], eax
	sub_31486A
	short loc_31306D
	eax, [ebp+arg_0]
	[ebp+arg_4]
	sub_314623
	short loc_31306D
	[ebp+arg_0], esi
	short loc_31308F

.oc_313066

; CODE XREF: sub 312FD8 ; sub 312FD8+59

- Way faster than emailing back and forth
- Some of us are usually on at ridiculous hours
 - basically a 24/7 channel

		loc 31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				8
		loc_31308C:			
				[ebp+var 4], ea	

Options of Last Resort

• Email us

MBE - 01/27/2015

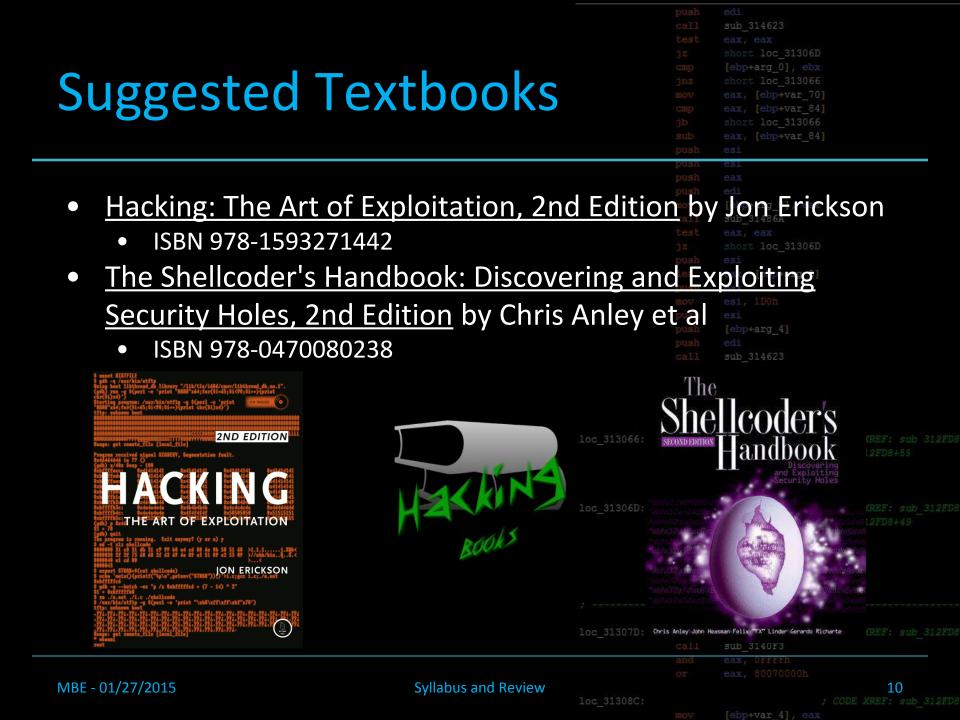
binexp_ta@cs.lists.rpi.edu

		sub_314623	
		short loc_31306D	
		<pre>[ebp+arg_0], ebx</pre>	
		short loc_313066	
		eax, [ebp+var_70]	
		eax, [ebp+var_84]	
		short loc_313066	
		eax, [ebp+var 84]	
	pusn	es1	
		[ebp+arg_0], eax	
		sub_31486A	
		short loc_31306D	
		eax, [ebp+arg_0]	
		[ebp+arg 4]	
		edi	
		short loc_31306D	
		[ebp+arg_0], esi	
		short loc 31308F	
066:			
		sub 31411B	
06D:			
		, sub_3140F3	
		eax, eax	
		short loc_31307D	
		sub_3140F3	
		short loc 31308C	
		50010 10C_31306C	
		sub 3140F3	
	and	eax, OFFFFh	
			9
			CODE VEEF: out 312EDS

Syllabus and Review

loc_31308C:

[ebn+var 4] eav



Grade Breakdown

- Labs 60%
 - 10 labs @ 6% each
 - Lab attendance is MANDATORY as labour to the submissions must be checked of the submission of the submissio
- Term Projects 40%
 - 2 Projects @ 20% each
 - Like a big lab, but over a few weeks

		loc_31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				11
		loc_31308C:			
				[ebp+var 4], eas	



An Atypical Class

- Designed and orchestrated by RPISEC (students)
- **Biggest RPISEC** class yet!
 - CSCI 4971 Secure Software Principles
 - CSCI 4972 / 6963 Malware Analysis
 - CSCI 4974 / 6974 Hardware Reverse Engineering

MBE	- 01/27/2015	Syllabus and Review	loc_31308C:			13 ; CODE XREF: sub_312FD8
			loc_31307D:	call	sub_3140F3	; CODE XREF: sub_312FD6
					eax, eax short loc_31307 sub_3140F3 short loc_313080	
•	We're not here to	mess arou	10c_31306D:		sub 3140F3	
					0Dh sub_31411B	

64 6f 6f 6d 2e 6c 79 6e 78 6a 65 72 2 RPTSEC: 2e 4c 65 7 RPTSEC: 67 65 65 66 64 2e 55 6e 69 78 2d

- Good to see lots of familiar faces!
- RPISEC meetings are <u>Friday 5-7 PM</u> in DCC 324
- Come learn other topics in computer security
 - Web hacking
 - Malware analysis
 - Reverse Engineering
 - Digital Forensics
 - So so much more

MBE - 01/27/2015

Meet people from industry, get internships/jobs

Syllabus and Review

Read more - http://rpis.ec

		short loc_31306D	
		<pre>[ebp+arg_0], ebx</pre>	
		short loc_313066	
		<pre>eax, [ebp+var_70]</pre>	
		eax, [ebp+var_84]	
		short loc_313066	
		eax, [ebp+var_84]	
	push	esi	
	pusn	031	
es!		[ebp+arg_0], eax	
		sub_31486A	
		eax, eax	
<u>' PM</u> in		324^{100}	
	lea	eax, [ebp+arg_0]	
_			
nputer	sec	curity	
	push	CSI	
		[ebp+arg_4]	
		sub_314623	
		short loc_31306D	
		[ebp+arg_0], esi	
		short loc_31308F	
loc 313066:			
100-112000.			
		sub 31411B	
loc 31306D:			
		sub 3140F3	
tintar	nch	inclich	C
<u>t mten</u>	ISI	ips/job	S
		short loc_31308C	
loc_31307D:			
	call	sub_3140F3	
		eax, OFFFFh	
			1.4
100 212090			14 : CODE XREF: sub 312FD8
loc_31308C:		[ebp+var 4] eax	

Lecture Overview

1. Syllabus

2. Course Overview

3. Review of Background Material

- a. Linux
- b. C

MBE - 01/27/2015

c. x86 Assembly

		sub_314623	
		short loc_31306D	
		[ebp+arg_0], ebx	
		short loc_313066	
		eax, [ebp+var_70	
		eax, [ebp+var_84	
		short loc_313066	
		eax, [ebp+var_84	
	pusn	651	
		<pre>[ebp+arg_0], eax</pre>	
		sub_31486A	
		short loc_31306D	
		<pre>eax, [ebp+arg_0]</pre>	
_			
al			
		[ebp+arg_4]	
		short loc_31306D	
		<pre>[ebp+arg_0], esi</pre>	
		short loc_31308F	
066:			
		sub_31411B	
06D:			
		sub_3140F3	
		short loc_31307D	
		sub_3140F3	
		short loc_31308C	
		sub 3140F3	
		eax, OFFFFh	
	and		
			15
			· CODE VEFF: out 212FD2

Sylla	bus	and	Revie
-------	-----	-----	-------

W

Course Terminology

- Machine
 - A computer, server, sometimes refers to the actual CPU
- Binary
 - An executable such as an .EXE, ELF, MachO or other code containers that run on a machine
 - Other names: program, application, service (sometimes)
- Malware
 - A malicious binary meant to persist on a machine such as a Rootkit or Remote Access Tool (RAT)

				sub_3140F3 short loc_31308C	
		, loc_31307D:		sub_3140F3	
			and	eax, OFFFFh	
1BE - 01/27/2015	Syllabus and Review	loc_31308C:		eax, 80070000h	16 ; CODE XREF: sub_312FD8

Course Terminology

Vulnerability

- A bug in a binary that can be leveraged by an exploit
- Exploit (as a noun)
 - Specially crafted data that utilizes vulnerabilities to force the binary into doing something unintended
 - By this definition, exploits are not explicitly malware
- Oday
 - A previously unknown or unpatched vulnerability that can be used by an exploit
 - An Oday can also be an exploit using the unpatched vuln

		loc_31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				17
		loc_31308C:			
				Johnwan //	

Premise For T	nis Class	push call test jz cmp jnz mov cmp jb sub push	edi sub_314623 eax, eax short loc_313 [ebp+arg_0], short loc_313 eax, [ebp+var eax, [ebp+var short loc_313 eax, [ebp+var esi	ebx 066 _70] _64] 066
			eax, eax short loc_313	
"Can we t			esi eax, [ebp+arg eax [ebp+arg_4] edi sub_314623	
programme	ers how t		short loc 313	6D Si OBF
	loc_3130			
 Pwn/Pwning 				
•	loc_3130			
 In security, pwn 	ing commonly r	eters	to, and eax	
vulnerability res	earch and explo	oit de	velopr	nent
	loc_3130	call	sub_3140F3	; CODE XREF: sub_312FD8
MBE - 01/27/2015	Syllabus and Review	and or 18C:		h 18 ; CODE XREF: sub_312FD8

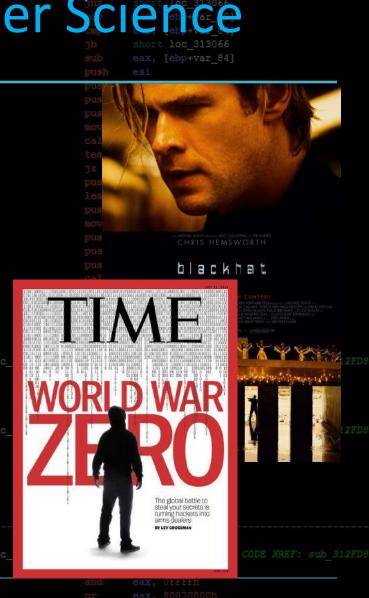
Goals for This Course

- This will be a very applied, hands on course
 - No data structures, algorithms, cryptography, or cyber policy
 - Every lecture after this you're expected to bring your laptop!
- We will cover technically challenging material rarely touched upon in other classes
- As an individual you will leave with all the skills necessary to perform vulnerability research, bypass modern security protections, and develop weaponized exploits

		loc 31307D:				
				sub_3140F3		
			and	eax, OFFFFh		
MBE - 01/27/2015	Syllabus and Review				19	
		loc_31308C:				
				John tran 41 eav		

"Dark Arts" of Computer Science

- Almost non-existent in academia
 - Taboo around offensive security
 - Rapidly evolving, very technical
- Why learn binary exploitation?
 - Can't defend against what you don't understand
 - Gain an intimate understanding of how programs *really* work
 - Fun, intriguing, rewarding problems
 - So few people know how to pwn
 - Exploding job market



Syllabus and Review

20

The Market for An Oday (2012

	call sub 31	1967
ADOBE READER	\$5,000-\$30,000	
MAC OSX	\$20,000-\$50,000	
ANDROID	\$30,000-\$60,000	
FLASH OR JAVA BROWSER PLUG-INS	\$40,000-\$100,000	
MICROSOFT WORD	\$50,000-\$100,000	
WINDOWS	\$60,000-\$120,000	
FIREFOX OR SAFARI	\$60,000-\$150,000	: CODE XREF: sub 31 ; sub 312FD8+59
CHROME OR INTERNET EXPLORER	\$80,000-\$200,000	
IOS	\$100,000-\$250,000	; CODE XREF: sub_31 ; sub 312FD8+49

2015? Double these numbers

		loc 31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				21
		loc_31308C:			
				Lohn man /1	

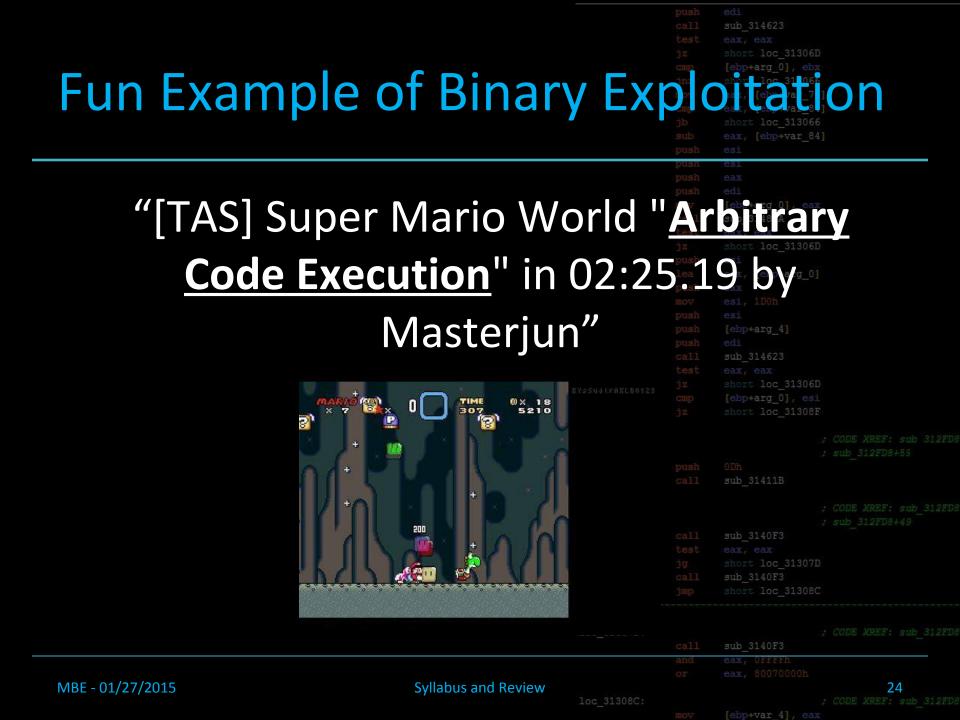
Underappr	eciated Wisd	<pre>push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], ebx on t loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi</pre>
	ogram simply der yourself der solle	<pre>push eax push edi mov [ebp+arg_0], eax call sub_31486A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push esi push fobp+arg_4] tebp+arg_4] tebp+arg_4] tebp+arg_4] tebp+arg_6], esi short loc_31306D iz short loc_31306D (cmp [ebp+arg_0], esi short loc_31308F ; sub_312FD8+55 push ODh call sub_31411B</pre>
	- Pro	of. ig ig imp short loc_31308C
MBE - 01/27/2015	امد_3130 Syllabus and Review	call sub_3140F3 and eax, 0FFFh or eax, 80070000h 222

More Than a Segfault

push	edi
	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	<pre>eax, [ebp+var_84]</pre>
	short loc_313066
	<pre>eax, [ebp+var_84]</pre>
pusn	.es1

- The right bugs (vulnerabilities) found in binaries can be used by exploits to hijack code execution
- Once code execution is achieved by an in a contraction is achieved by a number of the provided by a n
 - Gain privileged information 313066
 - Download or install malware
 - Steal data
 - Wreak any sort of havoc on the machine

		loc 31307D:			
				sub_3140F3	
			and	eax, OFFFFh	
MBE - 01/27/2015	Syllabus and Review				23
		loc_31308C:			



Events in Security & Exploitation 313060

- 1972 USAF Computer Security Technology Planning Study describes buffer overflows
- 1988 Morris Worm exploits use of gets() in finger daemon
- 1996 Aleph1 publishes "Smashing the Stack for Fun and Profit" in Phrack
- 2001 Code Red worm exploits a MS web server vulnerability to hit hundreds of thousands of computers
- 2004 Windows XP SP2 released, exploit mitigation era begins
- 2007 The first iPhone jailbreak is developed by GeoHot
- 2008-2010 Stuxnet employs four Windows Odays to spread through Iranian nuclear refinery control system networks

Syllabus and Review

					'n			
ь.	C.	<u> </u>	9		ω.	u.	6	

; CODE XREF: sub_

Course Roadmap

- We start off with the fundamentals required
 - Basic reverse engineering, memory corruption, classical exploitation
- Different classes of vulnerabilities are introduced and how they can be leveraged in exploitation
 - Stack smashes, format strings, signed/unsigned, Heap, UAF, etc

Syllabus and Re

- Modern exploit mitigations are introduced and how they can be bypassed in exploitation
 - DEP, ASLR, GS/Cookies,

MBE - 01/27/2015

			EdX, EdX	
			short loc_31306	
			[ebp+arg_0], eb	
			short loc_31306	
			eax, [ebp+var_7	
			eax, [ebp+var_8	
			short loc_31306	
			eax, [ebp+var_8	4]
		push	esi	
		pusn	651	
		pusn		
tals	require	ea	[ebp+arg_0], ea	
			sub_31486A	
α	rruntion	alace	ical avala	sitation
y CO	ուսբսօո,	CIGSS	ical explo	niation
			eax, [ebp+arg 0	
			eax	
			esi, 100h	
	re intro	duce	ed and h	
CS a		uuuu		
. •				
atic	n		sub 314623	
		JEL	short loc 21306	D
nea/	runsigned	I, Hea	ip, UAF, e	C
-	0	jz	short loc_31308	
	loc_313066:			
intr	oducod	and	how th	ov con
	ouuceu	anu		ey Call
			sub_31411B	
	loc_31306D:			
			sub_3140F3	
			eax, eax	
		call	short loc_31307 sub 3140F3	
			short loc_31308	
	loc 31307D:			
			sub 3140F3	
		and	eax, OFFFFh	
view				26
	loc 31308C:			; CODE XREF: sub 312FD8
			[ebp+var 4], ea	

Lecture Overview

- Syllabus
- Course Overview

• Review of Background Material

- Linux
- C
- x86 Assembly

		sub_314623	
		short loc_31306D	
		[ebp+arg_0], ebx	
		short loc_313066	
		<pre>eax, [ebp+var_70</pre>	
		eax, [ebp+var_84	
		short loc_313066	
		eax, [ebp+var_84	
	pusn	031	
		<pre>[ebp+arg_0], eax</pre>	
		sub_31486A	
		short loc 31306D	
		eax, [ebp+arg 0]	
		[ebp+arg_4] edi	
		sub_314623	
		short loc 31306D	
		[ebp+arg_0], esi	
		short loc 31308F	
66:			
		sub 31411B	
6D:			
		sub 3140F3	
		short loc_31307D	
		sub_3140F3	
		short loc 31308C	
		sub 3140F3	
	and	eax, OFFFFh	
			27
8C:			· CODE XREF: sub 312FDR

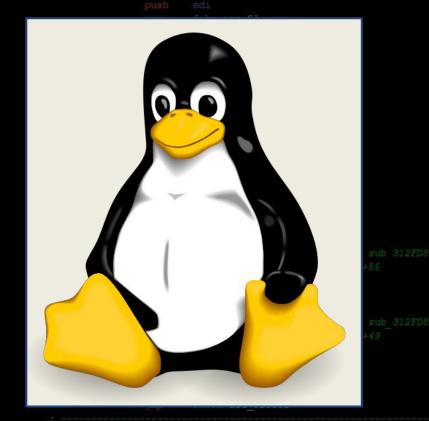
Syllabus and Review

Quick Linux Overview

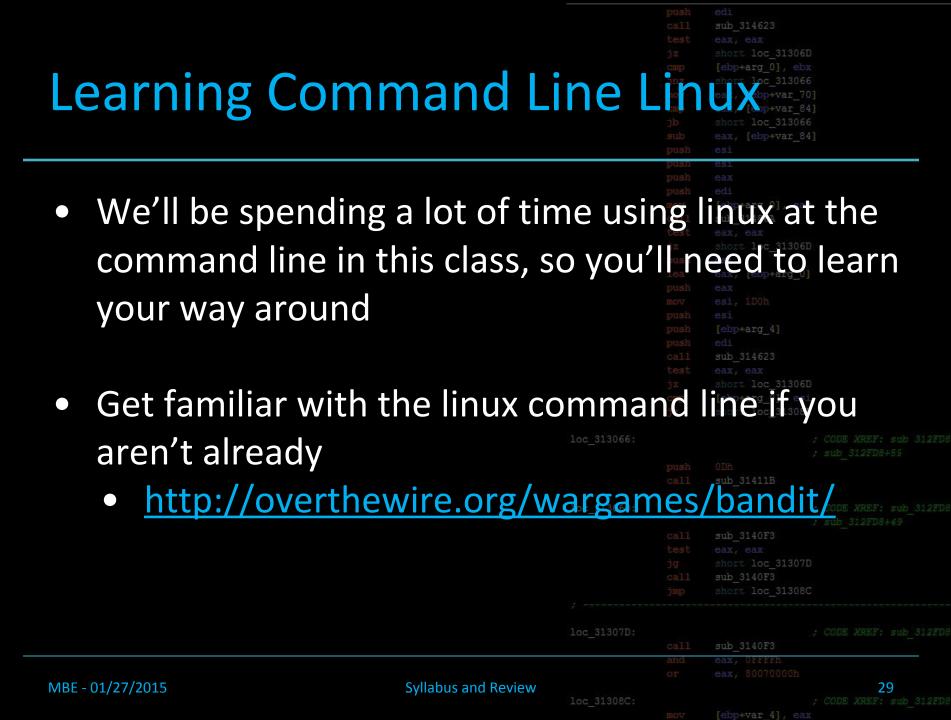
	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	eax, [ebp+var_70]
	<pre>eax, [ebp+var_84]</pre>
	short loc_313066
	eax, [ebp+var_84]
pusn	.es1

- UNIX-like open source kernel used by many open source operating systems distros
- Written in C and assembly
- ELF (Executable and Linkable Format) files for binaries
- We'll be teaching on Ubuntu 14.04 systems, but exploitation techniques are pretty universal

Syllabus and



	loc_31307D:			
			sub_3140F3	
		and	eax, OFFFFh	
Review				28
	loc_31308C:			
			Lebnation (1)	



Basic Command Line	Usa	push call test jz cmp jb sub push	edi sub_314623 eax, eax short loc_3130 [ebp+arg_0], e short loc_3130 eax, [ebp+var_ short loc_3130 eax, [ebp+var_ short loc_3130 eax, [ebp+var_	bx 66 70] 84] 66
 ls List directory contents cd [path] change directory "" = previous pwd Print working directory man [command] 	loc_313066:	push push push mov call test jz push lea push push push call test jz cmp jz	esi eax edi [ebp+arg_0], e sub_31486A eax, eax short loc_3130 esi eax, [ebp+arg_ eax esi, 1D0h esi [ebp+arg_4] edi sub_314623 eax, eax short loc_3130 [ebp+arg_0], e short loc_3130	
 Manual for command 			0Dh sub_31411B	
• apropos [whatever]	loc_31306D:		sub_3140F3	
 Get info on commands/man pages that 	at might o		hatever sub_3140F3 short loc_3130	
MBE - 01/27/2015 Syllabus and Review	loc_31307D: loc_31308C:	call and or	sub_3140F3 eax, OFFFFh eax, 80070000h	; CODE XREF: sub_312FD8 30 ; CODE XREF: sub_312FD8

Working With Files

- cat [file]
 - Print the file contents on your terminal
- less [file]
 - Like cat, but paged, good for long documents
- mv [file1] [file2]
 - Move file1 to file2, removing file1 and overwriting file2 if it exists

Syllabus and Review

- cp [file1] [file2]
 - Copy file1 to file2, overwriting file2 if it exists
- rm [file]

MBE - 01/27/2015

- Deletes file
- nano / vim / emacs
 - Command line text editors

		sub 314623	
		short loc_31306	
		[ebp+arg_0], ebs	
		short loc 31306	
		eax, [ebp+var_70	
		eax, [ebp+var_84	
		short loc_31306	
		eax, [ebp+var_84	
	pusn	631	
		[ebp+arg_0], eau	
		sub_31486A	
		short loc_31306	
		eax, [ebp+arg_0]	
nents			
		[ebp+arg_4]	
		sub_314623	
•	test	lear ear	• .
verwrit	ting f	ile2 if it ex	kists
	CIED	[ebp+arg_0], esi	
		short loc_313081	
exists			
exists			
		sub_31411B	
oc_31306D:			
		sub_3140F3	
		short loc_313071	
		sub_3140F3	
		short loc_313080	
oc 31307D:			
-aradin.		sub 3140F3	
		eax, OFFFFh eax, 80070000h	
			31
oc 31308C:			; CODE XREF: sub 312FD8

Piping Program Input	: / Οι	push call test jz cmp rat jb sub push	short loc_313066 eax, [ebp+var_84] esi	
 Pipes - " " Take output of one program, send it as in \$ echo "hello" cowsay 	put to anot	push push mov call her jz push lea push mov	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0] eax esi, 1D0h	
< hello >			esi [ebp+arg_4] edi	
 \ ^^			<pre>sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi short loc_31308F</pre>	
\ (oo) \ () \	loc_313066:			
• \$ echo "hello" cowsay grep "	" 100_31306D: "hello"			
< hello >	loc_31307D:	call	sub_3140F3	: CODE XREF: sub_312FD8
MBE - 01/27/2015 Syllabus and Review	loc_31308C:	and	eax, 0FFFFh eax, 80070000h	32 : CODE XREF: sub_312FD8

Lecture Overview

- Syllabus
- Course Overview

Review of Background Material

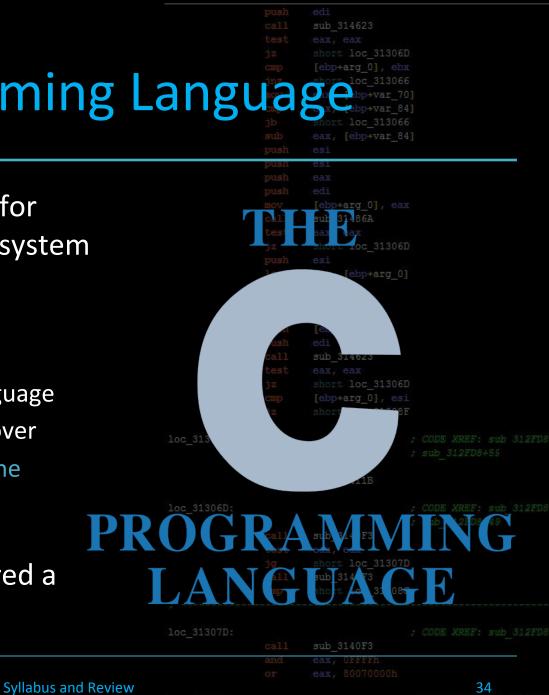
- Linux
- C
- x86 Assembly

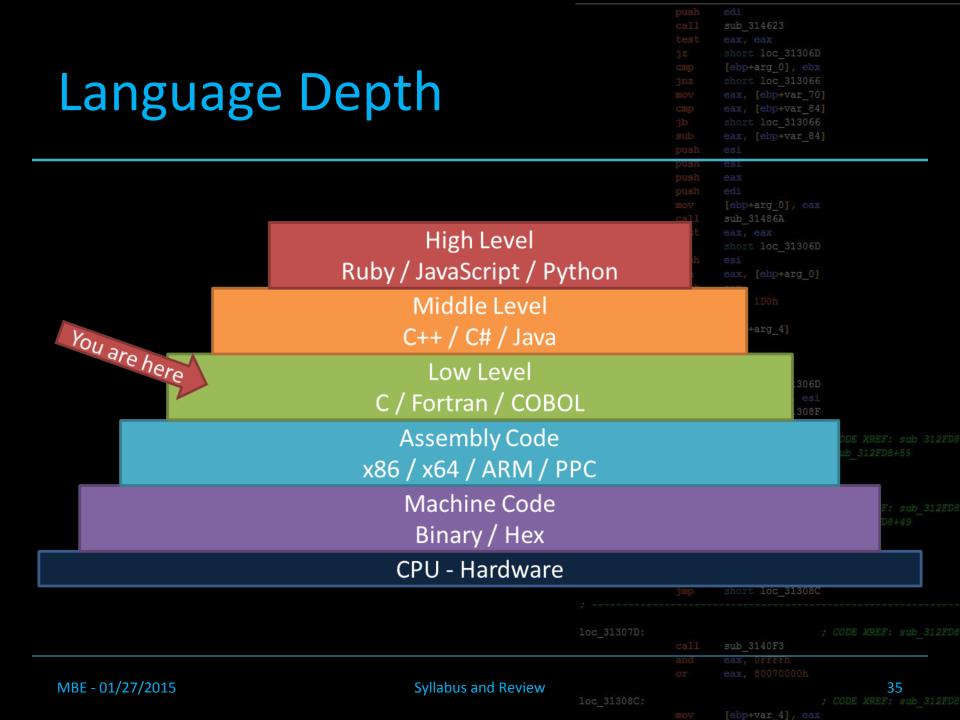
		sub_314623	
		short loc_31306	
		[ebp+arg_0], eb	
		short loc_31306	
		<pre>eax, [ebp+var_7</pre>	
		short loc_31306	
		eax, [ebp+var_8	4]
	pusn	CSI	
		<pre>[ebp+arg_0], ea</pre>	
		sub_31486A	
		short loc_31306	
		eax, [ebp+arg_0)]
rial			
		[ebp+arg_4]	
		sub_314623	
		short loc_31306	D.
		[ebp+arg_0], es	
		short loc_31308	
0100CC			
:_313066:			
		0Dh	
		sub_31411B	
212060-			
_31306D:			
		aub 2140F2	
		eax, eax	
		short loc_31307 sub 3140F3	
		short loc 31308	
		Short loc_31308	
: 31307D:			
		sub 3140F3	
		eax, OFFFTh	
	and or		
			33
: 31308C:			· CODE XREF: sub 312FD8

Syllabus and Review

The C Programming Languag

- Designed in 1969-1972 for writing UNIX operating system
- Imperative systems programming language
 - Very fast, compiled language
 - Extremely fine control over memory and the machine
- Compared to modern languages, C is considered a 'low level' language





Hello World! - C Sou	irce	push call test jz cmp jnz mov cmp jb sub sub	edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70 eax, [ebp+var_84 short loc_313066 eax, [ebp+var_84 esi	
<pre>#include <stdio.h></stdio.h></pre>			esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0]	
<pre>int main(int argc, char {</pre>	* argv		eax esi, 1D0h esi [ebp+arg_4] edi sub_314623	
printf("Hello World!\r	1");		<pre>eax, eax short loc_31306D [ebp+arg_0], esi short loc_31308F</pre>	
return 0;	loc_313066:			
	loc_31306D:		<pre>sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C</pre>	
MBE - 01/27/2015 Syllabus and Revie	loc_31307D:	call and or mov	sub_3140F3 eax, OFFFFh eax, 80070000h	; CODE XREF: sub_312FD8 36 ; CODE XREF: sub_312FD8

Hello World! - Com	piling/	push call test jz cmp cmp jb sub push	short loc_313066 eax, [ebp+var_84] esi
<pre>\$ gcc helloworld.c \$./helloworld</pre>	-o he	push push mov call test jz push lea push mov push call test call test jz cmp jz	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0] eax esi, 1D0h esi sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi short loc_31308F
Hello World!	loc_313066:		; CODE XREF: sub 312FD ; sub_312FD8+55 ODh sub_31411B
	loc_31306D: ;		; CODE XREF: sub_312FD0 ; sub_312FD8+49 sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C
MBE - 01/27/2015 Syllabus and Re	loc_31307D: eview loc_31308C:	call and or mov	; CODE XREF: sub_312FD; sub_3140F3 eax, 0rffrh eax, 80070000h 37 ; CODE XREF: sub_312FD; [ebp+var_4], eax

Basic Memory	Manip	ulati	push call test jz cmp fo jb sub push	<pre>short loc_313066 eax, [ebp+var_84] esi</pre>	
<pre>int i = 0; char * message = "hello char * buffer = (char *</pre>				esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0]	
<pre>if(buffer == NULL) return 1;</pre>				<pre>eax esi, 1D0h esi [ebp+arg_4] edi sub_314623 eax, eax short loc 31306D</pre>	
<pre>strncpy(buffer, message buffer, log</pre>	e, 5);			<pre>[ebp+arg_0], esi short loc_31308F</pre>	
<pre>buffer[5] = '\n'; buffer[6] = '\0';</pre>		loc_313066:			
<pre>for(i = 0; i < 10; i++) printf("%s", buffer</pre>	·);	loc_31306D:		<pre>sub_3140F3 eax, eax short loc_31307D sub_3140F3</pre>	
free(buffer);		;		short loc_31308C	
MBE - 01/27/2015	Syllabus and Review	loc_31308C:	and or mov	eax, OFFFFh eax, 80070000h [ebp+var 4], eax	38 : CODE XREF: sub_312FD8

				sub_314623	
				short loc_31306D	
— • • •				[ebp+arg_0], ebx	
Running It				short loc_313066	
конния п				eax, [ebp+var_70	
				eax, [ebp+var_84	
				short loc_313066	
				eax, [ebp+var_84	
			pusn	651	
\$ gcc basic.c	-o hacic -	$-\alpha + d - \alpha$	ΩτιΩ	Qebp+arg 0], eax	
\$ gcc basic.c	U DASIC	stu-y.	пцэ	sub 31486A	
\$./basic				short loc_31306D	
\$./basic					
				<pre>eax, [ebp+arg_0]</pre>	
hello					
				esi John ong (1	
				[ebp+arg_4] edi	
hello				sub_314623	
				eax, eax	
				short loc_31306D	
hello				[ebp+arg_0], esi	
				short loc_31308F	
hello		loc_313066:			
110 1 10					
hello				sub_31411B	
		loc_31306D:			
				sub 3140F3	
• • •				eax, eax	
				short loc_31307D	
				sub_3140F3	
				short loc_31308C	
		loc_31307D:			
			call	sub_3140F3	
			and	eax, OFFFFh	
	Culloburg and Day				20
MBE - 01/27/2015	Syllabus and Review	loc_31308C:			39 ; CODE XREF: sub_312F
		100-212000:		[ebp+var 4], eax	
				Combernar -11 Cav	

What's your name	9?	push call jz cmp jnz mov cmp jb sub push	edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84] esi	
<pre>#include <stdio.h> #include <unistd.h></unistd.h></stdio.h></pre>		push push mov call test jz push lea push mov push	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0] eax esi, 1D0h esi	
<pre>int main(int argc, char buffer[10]</pre>) ;	test jz cmp	<pre>[ebp+arg_4] edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi hist loc_31308F</pre>	
<pre>printf("What's your read(STDIN_FILENO,]</pre>	buffer, 10			XREF: sub 312FD ; sub 12FD8+55 ; CODE XREF: sub 312FD ; sub_312FD8+49
<pre>printf("Hello %s\n" return 0; }</pre>			<pre>sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C</pre>	
MBE - 01/27/2015 Syllabus an		call and or	sub_3140F3 eax, OFFFFh eax, 20070000h	40 ; CODE XREF: sub_312FD

Hello ALEX 1234 ??

\$ gcc name.c -o name \$./name What's your name? ALEX 1234 ABCD Hello ALEX 1234 ??

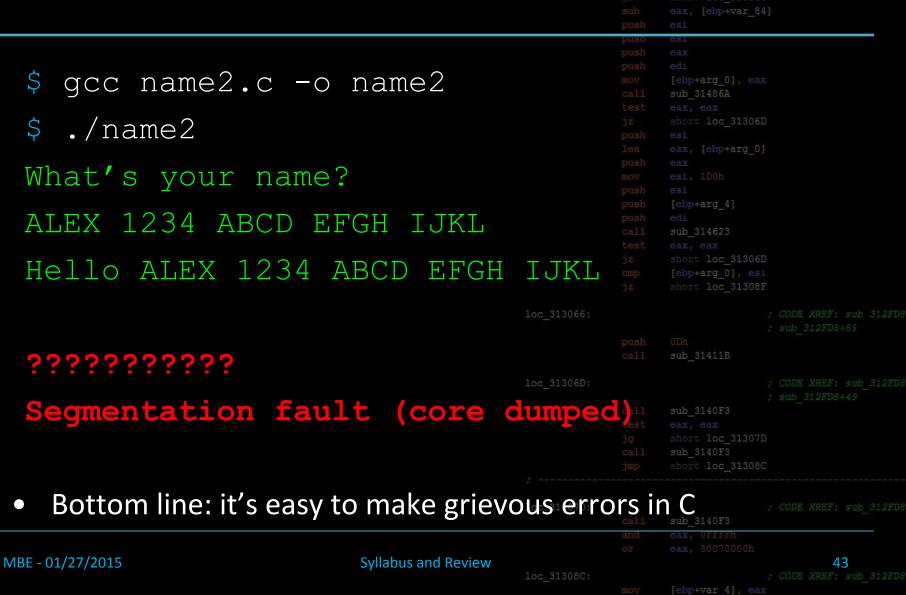
Syllabus and Review

MBE - 01/27/2015

		edi
		sub_314623
		short loc_31306D
		[ebp+arg_0], ebx
		short loc_313066
		eax, [ebp+var_70]
		eax, [ebp+var_84]
		short loc_313066
		eax, [ebp+var_84]
	push	esi
	pusn	007
		edi Ishnuarg 01 sar
		[ebp+arg_0], eax
		sub_31486A
		eax, eax
		short loc_31306D
		esi
		eax, [ebp+arg_0]
		esi [ebp+arg 4]
		edi
		sub_314623
		eax, eax
		short loc_31306D
		[ebp+arg_0], esi
		short loc_31308F
3066:		
		sub_31411B
.306D:		
		sub_3140F3
		short loc 31307D
		sub_3140F3
		short loc_31308C
		sub_3140F3
	and	eax, OFFFFh
		41
.308C:		; CODE XREF: sub 312FD8
		[ebp+var 4], eax

What's your name?	jb sub pusl	sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84] short loc_313066	
<pre>#include <stdio.h> #include <unistd.h></unistd.h></stdio.h></pre>	pus pus mov cal tes jz pus lea pus mov pus	<pre>i eax edi [ebp+arg_0], eax sub_31486A c eax, eax short loc_31306D esi eax, [ebp+arg_0] eax esi, 1D0h esi</pre>	
<pre>int main(int argc, char char buffer[10] = {0}</pre>	test		
printf("What's your n read(STDIN_FILENO, bu		0Dh sub_31411B	
<pre>printf("Hello %s\n", return 0;</pre>		sub_3140F3 eax, eax short loc_31307D	
} MBE - 01/27/2015 Syllabus and Rev	loc_31307D: call and or /iew loc_31308C:	sub_3140F3 eax, 0FFFFh eax, 80070000h	CODE XREF: sub_312FD8

Crash!



So If C Scared You...

MBE

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
pusn	esi

ub 314623

- If you're in this class, we expect you to already know some basic C from CompOrg, CANOS, OpSys, or NetProg
- Otherwise, review C programming ASAP
 - "Hacking: The Art of Exploitation", chapter 0x200

01/27/2015	Syllabus and Review	loc 31308C:	call and or	sub_3140F3 eax, 0FFFFh eax, 80070000h	44 ; CODE XREF: sub 312FD8
		loc_31307D:			
				<pre>sub_3140F3 eax, eax short loc_31307I sub_3140F3 short loc_313080</pre>	
		loc_31306D:			

Lecture Overview

- Syllabus
- Course Overview
- Review of Background Material
 - Linux
 - C
 - x86 Assembly

		sub_314623	
		short loc_31306D	
		[ebp+arg_0], ebx	
		short loc_313066	
		<pre>eax, [ebp+var_70]</pre>	
		eax, [ebp+var_84]	
		short loc_313066	
		eax, [ebp+var_84]	
	pusn	esi	
		[ebp+arg 0], eax	
		sub_31486A	
		short loc 31306D	
		eax, [ebp+arg 0]	
al			
aı		[ebp+arg 4]	
		sub_314623	
		short loc_31306D	
		[ebp+arg_0], esi	
		short loc 31308F	
3066:			
		sub_31411B	
306D:			
		sub_3140F3	
		short loc_31307D	
		sub 3140F3	
		short loc 31308C	
307D:			
		sub 3140F3	
	and	eax, OFFFFh	
			45

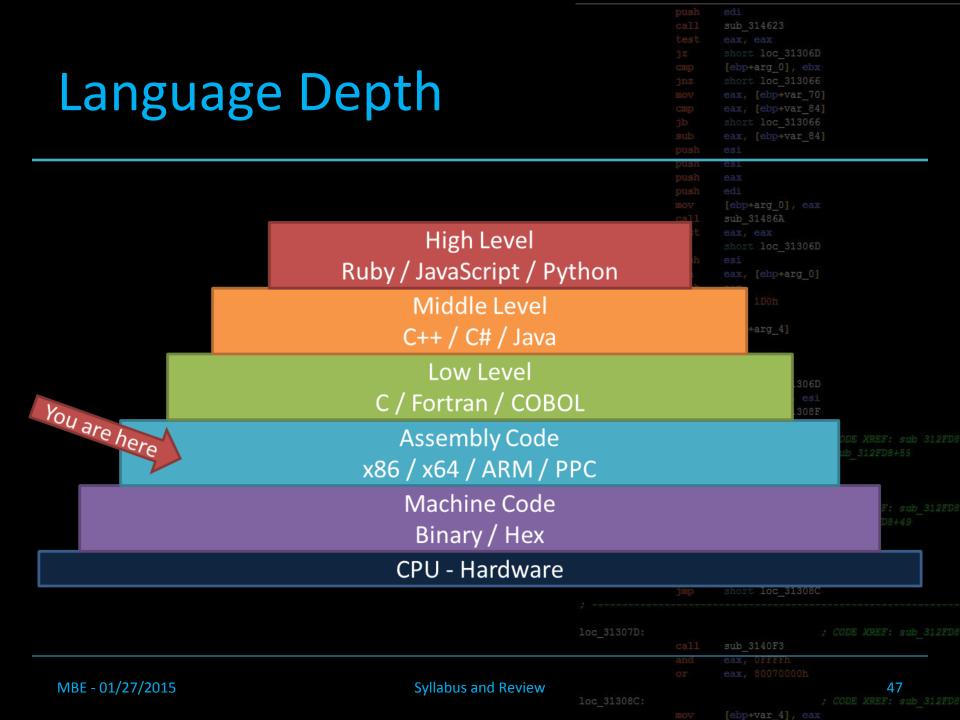
Syllabus and Review

x86 Assembly

- An assembly instruction set igodotintroduced in 1978 by Intel
 - 1978 16bit igodol
 - 1985 32bit \bullet
 - 2001 64bit (Itanium) \bullet
 - 2003 64bit (AMD64) ightarrow
- Overwrought CISC, a total igodolplayground for exploitation
- As low level as we'll go

		sub_314623	
		eax, eax short loc 31306D	
		[ebp+arg_0], ebx	
		short loc 313066	
		eax, [ebp+var_70]	
		eax, [ebp+var_64]	
		short loc_313066	
		eax, [ebp+var_84]	
	push	esi	
	pusn	651	
imo imo	US1, 1	PDO	
jmp	loc_80560	688	
		; CODE XI	
cmp	esp+48Cl	h+var_494], 45h 🛊	
jz	loc_80560	CA 9	
mov		p+4BCh+var_478]	
mov		h+var_470], 1	
mov	esi, [eax		
jmp	10c_8056I	000	
		- 0005 - 10	
		; CODE XI	
стр	[esp+4BCl	; CODE XI h+var_494], 45h ;	
cmp jz	[esp+48C loc_8056(h+var_494], 45h	;
	10C_80560	h+var_494], 45h ; CA0	;
jz	ioc_80560 esi, [esp	h+var_494], 45h ; CA0 p+4BCh+var_478]	REF: sub 3121
jz mov mov	loc_80560 esi, [esp ebx, 9249	h+var_494], 45h CAO p+4BCh+var_478] 92493h	REF: sub 3121
jz mov mov mov	loc_80560 esi, [esp ebx, 9249 eax, [esp	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h]	REF: sub 3125 2FD8+55
jz mov mov mov lea	loc_80560 esi, [esj ebx, 9249 eax, [est ecx, [est	h+var_494], 45h CAØ p+4BCh+var_478] 92493h i+18h] x+6]	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov nov lea mov	loc_80560 esi, [es] ebx, 9249 eax, [esi ecx, [eax [esp+48C]	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h]	REF: sub 3125 2FD8+55
jz mov mov nov lea mov mov	loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+4BC] eax, ecx	h+var_494], 45h CAØ p+4BCh+var_478] 92493h i+18h] x+6]	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov lea mov mov imul	loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+48C] eax, ecx ebx	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov nov lea mov mov	<pre>loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+4BC0 eax, ecx ebx eax, [edx</pre>	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov lea mov mov imul	loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+48C] eax, ecx ebx	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov lea mov mov imul lea	<pre>loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+4BC0 eax, ecx ebx eax, [edx</pre>	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 3125 2FD8+55 REF: sub_3125
jz mov mov lea mov mov imul lea mov sar	loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+48C] eax, ecx ebx eax, [edx edx, ecx edx, ecx	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 3128 2FD8+55 REF: sub 3128 2FD8+49
jz mov mov lea mov mov imul lea mov	<pre>loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+4BC0 eax, ecx ebx eax, [edx edx, ecx edx, 1Fh eax, 2</pre>	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax x+ecx]	REF: sub 3128 2FD8+55 REF: sub 3128 2FD8+49
jz mov mov lea mov mov imul lea mov sar	loc_80560 esi, [es] ebx, 9249 eax, [es] ecx, [eax [esp+48C] eax, ecx ebx eax, [edx edx, ecx edx, ecx	h+var_494], 45h CA0 p+4BCh+var_478] 92493h i+18h] x+6] h+var_48C], eax	REF: sub 312F 2FD8+55 REF: sub_312F

Syllabus and Review



Pulling Back the Curtain

	sub_314623
	short loc_31306D
	[ebp+arg_0], ebx
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	eax, [ebp+var_84]
	short loc_313066
	eax, [ebp+var_84]
ousn	631
	eax

	.text:00428DC8					1oc_428DC8		; CODE XREF: sub_428CE0+AB†j	
	.text:00428DC8							; sub_428CE0+C1†j	
! ≯• ∣	.text:00428DC8	8A 8	35 2B	04 0	0 00		mov	al, [ebp+42Bh]	
¦ •	.text:00428DCE	84 0	0				test	al, al	
-	.text:00428DD0		35 5A	02 0	0 00		jnz	loc_429030	
i 🍨	.text:00428DD6	8B 8	85 18	04 0	0 00		mov	eax, [ebp+418h]	
¦ •	.text:00428DDC	8B 6	90				mov	eax, [eax]	
i 🍨	.text:00428DDE	89 4	44 24	10			mov	[esp+494h+var_484], eax	
-	.text:00428DE2	8A 8	3D 2A	04 0	0 00		mov	cl, [ebp+42Ah]	
i •	.text:00428DE8	84 0	:9				test	cl, cl	
¦ •	.text:00428DEA	74 5	53				jz	short loc_428E3F	
i i 🎴	.text:00428DEC	3B 8	35 18	04 0	0 00		cmp	eax, [ebp+418h]	
¦	.text:00428DF2		34 35	02 0	0 00		jz	1oc_42902D	
	.text:00428DF8								
	.text:00428DF8					1oc_428DF8:		; CODE XREF: sub_428CE0+158↓j	
i i * •	.text:00428DF8	8B 8	38 10		0 00		mov	ecx, [eax+ <mark>810h</mark>]	
	.text:00428DFE	8D 5	54 24	54			lea	edx, [esp+494h+var_440]	
1 1 1 🖷 🎴	.text:00428E02	8D 4	44 24	78			lea	eax, [esp+494h+var_41C]	
	.text:00428E06	89 4	4C 24	54			mov	[esp+494h+var_440], ecx	
1 1 1 1	.text:00428E0A						push	edx	
	.text:00428E0B	50					push	eax	
1 1 1 1	.text:00428E0C				0 00		lea	ecx, [edi+12B4h]	
	.text:00428E12	89 7	74 24	60			mov	[esp+49Ch+var_43C], esi	
	.text:00428E16	E8 F	5 04	00 0			call	sub_429310	
	1 1 00100545	00.0	2.0						

1 sub 3140F3

"... there's way too much information to decode the Matrix. You get used to it, though. Your brain does the translating. I don't even see the code. All I see is blonde, brunette, redhead." -Cypher, The Matrix

, ,	, in the second s		loc_31307D:				
					sub_3140F3		
				and	eax, OFFFFh		
MBE - 01/27/2015		Syllabus and Review				48	
			loc_31308C:				

x86 Assembly Syntax

- push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], ebx jnz short loc_313066 mov eax, [ebp+var_70] cmp eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi push esi
- All assembly languages are made up of instruction sets
- Instructions are generally simple arithmetic operations that take registers or constant values as arguments
 - Also called Operands, OpCode, Op(s), mnemonic
- Intel syntax: operand destination, source
 - mov eax, 5
- AT&T syntax: operand source, destination
 - mov \$5, eax
- We'll be using the Intel syntax in this class

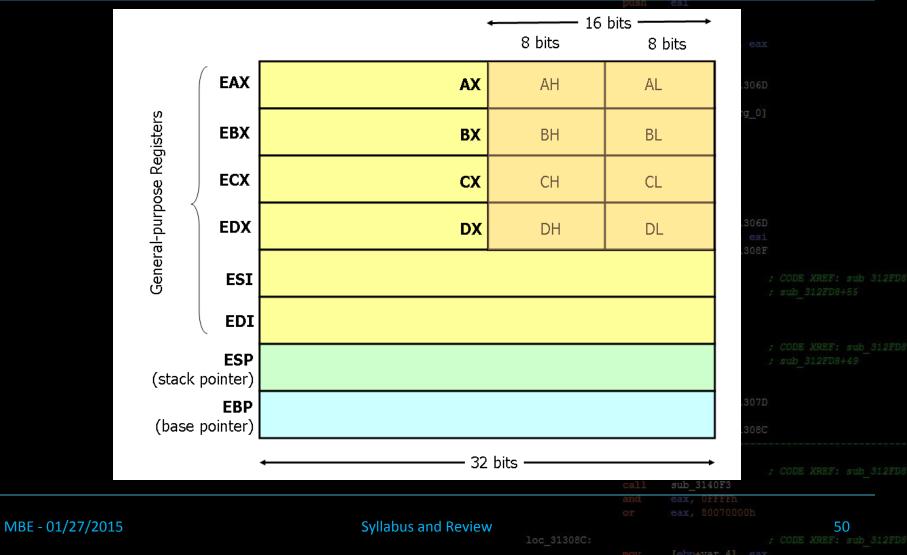
Call	sub_31486A	
op	erations	that
push	esi	
let c	<pre>eax, [ebp+arg_0</pre>	
nts		
push		
:S _{sn}	[ebp+arg_4]	
	sub_314623	
	short loc_31306	
	<pre>[ebp+arg_0], es</pre>	
	short loc_31308	
	sub_31411B	
	sub_3140F3	
	short loc_31307	
	sub_3140F3	
	short loc_31308	C
	sub_3140F3	
and	eax, OFFFFh	
		49

Syllabus and Review

; CODE XRE [ebp+var 4]. eax

x86 Register Diagram

push	edi
	sub_314623
	short loc_31306D
	<pre>[ebp+arg_0], ebx</pre>
	short loc_313066
	<pre>eax, [ebp+var_70]</pre>
	<pre>eax, [ebp+var_84]</pre>
	short loc_313066
	<pre>eax, [ebp+var_84]</pre>



Important Registers

- pushedicallsub_314623testeax, eaxjzshort loc_31306Dcmp[ebp+arg_0], ebxjnzshort loc_313066moveax, [ebp+var_70]cmpeax, [ebp+var_84]jbshort loc_313066subeax, [ebp+var_84]pushesipushesi
- EAX EBX ECX EDX General purpose registers
- ESP Stack pointer, "top" of the current stack frame (lower memory)
- EBP Base pointer, "bottom" of the current stack frame (higher memory)
- EIP Instruction pointer, pointer to the *next* instruction to be executed by the CPU
- EFLAGS stores flag bits
 - ZF zero flag, set when result of an operation equals zero
 - CF carry flag, set when the result of an operation is too large/small
 - SF sign flag, set when the result of an operation is negative

	0	Ċ,		loc_31307D:		0	
						sub_3140F3	
					and	eax, OFFFFh	
MBE - 01/27/2015			Syllabus and Review				51
				loc 31308C:			
						Lehnevar 41 e	

Moving Data

mov ebx, eax

Move the value in eax to ebx

Move 0xDEADBEEF into eax

mov ecx, DWORD PTR [edx]

mov eax, OxDEADBEEF

mov edx, DWORD PTR [0x41424344] Move the 4-byte value at address 0x41424344 into edx Move the 4-byte value at the address in edx, into ecx mov eax, DWORD PTR [ecx+esi*8] Move the value at the address ecx+esi*8 into eax 52

MB	F -	01	27	/20)15
	_	<u> </u>	,		

igodol

igodol

igodot

igodol

igodol

Syllabus and Review

Arithmetic Op	erations	push call test jz cmp jnz mov cmp jb sub push	edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70] eax, [ebp+var_84] short loc_313066 eax, [ebp+var_84] esi	
 sub edx, 0x11 edx = edx - 0x11; add eax, ebx 	<pre>// subtracts 0x11 fron</pre>	push push mov call n edx ^{test} push lea push	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi eax, [ebp+arg_0] eax	
• eax = eax + ebx;	// add eax and ebx, st		alue in eax	
 inc edx edx++; dec ebx 	// increments edx		<pre>[ebp+arg_4] edi sub_314623 eax, eax short loc_31306D [ebp+arg_0], esi short loc_31308F</pre>	
 ebx; xor eax, eax 	// decrements ebx	66: push call		
•	// bitwise xor eax wit	n itself (Zeros eax) sub_3140F3 eax, eax short loc_31307D sub 3140F3	
• edx = edx 0x1337;	// bitwise or edx with	0x1337	short loc_31308C	
	loc_3130	7D: call and	sub_3140F3 eax, OFFFFh	CODE XREF: sub_312FD8
MBE - 01/27/2015	Syllabus and Review			53 : CODE XREF: sub_312FD8

Some Conditi	onal Jur	nps	test jz cmp jnz mov cmp jb sub push	<pre>sub_314623 eax, eax short loc_31306D [ebp+arg_0], ebx short loc_313066 eax, [ebp+var_70 eax, [ebp+var_84 short loc_313066 eax, [ebp+var_84 esi</pre>	
			pusn push	esi eax	
• jz \$LOC				edi [ebp+arg_0], eax	
				sub_31486A eax, eax	
 Jump to \$LOC if ZF = 	1			short loc_31306D	
• jnz \$LOC				esi eax, [ebp+arg_0]	
JIIZ JLUC					
 Jump to \$LOC if ZF = 	0				
				[ebp+arg_4]	
• jg \$LOC				edi sub_314623	
 Jump to \$LOC if the r 	esult of a compa	rison is th	ne de	stination i	S
greater than the sour				<pre>[ebp+arg_0], esi short loc_31308F</pre>	
		loc_313066:			
				0Dh sub_31411B	
		loc_31306D:			
				sub_3140F3	
				eax, eax short loc 31307D	
				short loc_31308C	
		loc_31307D:			
		72	call	sub_3140F3	
MBE - 01/27/2015	Syllabus and Review	loc_31308C:			54 ; CODE XREF: sub_312FD8
				[ebp+var 4], eax	

Stack Manipulation	<pre>push edi call sub_314623 test ear, ear jz short loc_31306D cmp [ebp+arg_0], ebx jnz short loc_313066 mov ear, [ebp+var_70] cmp ear, [ebp+var_84] jb short loc_313066 sub ear, [ebp+var_84] push esi</pre>
 push ebx Subtract 4 from the stack pointer to me (zero,) and copy the value in EBX on to sub esp, 4 mov DWORD PTR [esp], ebx 	push esi
• pop ebx	jz short loc_31308F loc_313066: ; CODE XREF: sub 312FD8
 Copy the value off the top of the stack stack pointer to move it towards highe 	call sub_31411B
mov ebx, DWORD PTR [esp]	call sub_3140F3 test eax, eax

MBE - 02

	•			Ċ	106_313000.			; sub_312FD8+49
mov	ebx,	DWORD	PTR	[esp]			sub_3140F3 eax, eax	
add	esp,	4					short loc_31307 sub_3140F3	
					loc_31307D:		sub_3140F3	
						and or	eax, OFFFFh eax, 80070000h	
1/27/2015			Syllab	us and Review	loc 31308C:			55 ; CODE XREF; sub 312FD8

Calling / Returning	push edd call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], ebx jnz short loc_313066 mov eax, [ebp+var_70] cmp eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi	
 call some_function Calls the code at some_function. We 	e need to push the return	
<pre>address onto the stack, then branch push eip mov eip, some_function </pre>	mov esi, 1D0h push esi push [ebp+arg_4]	
 Used to return from a function call. I pop eip ; not actual? nop nop 'no operation' - does nothing 	Pops the top of the stack to eip call sub_31411B ly valid loc_31306D: call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C ; CODE XREF: sub_312E	
MBE - 01/27/2015 Syllabus and Review	call sub_3140F3 and eax, 0FFFFh or eax, 80070000h / 56 loc_31308C: ; CODE XREF: sub_312F	

Basic x86

N

0x08048	8624: "YOLOSWAG\0"
mov	ebx, 0x08048624
mov	eax, 0
LOOPY:	
mov	cl, BYTE PTR [ebx]
cmp	cl, O
jz	end
inc	eax
inc	ebx

BE - 01/27/2015	Syllabus and Review				57
			and	eax, OFFFFh	
		loc_31307D:		sub_3140F3	
ret					
end:				short loc_31307 sub_3140F3 short loc_31308	
jmp LOOPY				sub_3140F3 eax, eax	
inc ebx		loc_31306D:			
				sub_31411B	

631

Basic x86	pushedicallsub_314623testeax, eaxjzshort loc_31306Dcmp[ebp+arg_0], ebxjnzshort loc_313066moveax, [ebp+var_70]cmpeax, [ebp+var_84]jbshort loc_313066subeax, [ebp+var_84]jbshort loc_313066subeax, [ebp+var_84]	
	; 9 bytes of string[edata eax ; char * ebx = "YOLOSWAG.0";	
mov eax, 0 LOOPY:	; set eax to 0 lea eax, [ebp+arg_0] ; label, top of publoco	
mov cl, BYTE PTR [ebx] cmp cl, 0	[]; char cl = $*ebx_{11}^{push}$ edi sub_314623 ; is cl 0? (eg $\sqrt[1]{2} 0^{"s}$)ort loc_31306D [ebp+arg_0], esi	
jz end inc eax	; if cl was 0, go to end loc_313066: ; eax++; (counter for length) ¹²⁷	
inc ebx jmp LOOPY	; $ebx++$; $([ebx]] = "Y", "O" \dots " 0"; CODE XRE; go to LOOPY call sub_3140F3 test eax, eax$) 2: sub_312FD D8+49
end: ret		
MBE - 01/27/2015 Syll	call sub_3140F3 and eax, OFFFFn or eax, 80070000h /llabus and Review loc_31308C: ; CODE XRE mov [ebp+var_4], eax	58 F: sub_312FD

Human Decompil	jb short loc_313066 sub eax, [ebp+var_84] push esi
0x08048624: "YOLOSWAG\0" mov ebx, 0x08048624	<pre>push esi push eax push edi mov [ebp+arg_0], eax call sub_31486A char * word_jz wiyoft.ogSWAG"; int len = 0!ea eax, [ebp+arg_0]</pre>
mov eax, 0 LOOPY: mov cl, BYTE PTR [ebx]	<pre>int len = Opa eax, [ebp+arg_0] pash eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623</pre>
cmp cl, 0 jz end	while (*word !=bor(01)c_31306D [ebp+arg_0], esi short loc_31308F { loc_313066: ; CODE XREF: sub 312ED8
inc eax inc ebx	<pre>len++; push ODh call sub_31411B WOrd++; ; CODE XREF: sub_312FD8</pre>
jmp LOOPY end:	; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C
ret	return len; loc_31307D: ; CODE XREF: sub_312FD8 call sub_3140F3 and eax, OFFFh
MBE - 01/27/2015 Syllabus a	and Review 59 loc_31308C: ; CODE XREF: sub_312FD8

Additional Material

Related Readings: ightarrow

MBE - 01

- Hacking: The Art of Exploitation
 - Chapter 0x200: Programming C programming and GDB \mathbf{O}
- Practical Reverse Engineering (Dang et al)
 - Chapter 1 (x86)

- [ebp+arg_0], es: Get familiar with the linux command line if you alreadv

27/	/2015	Syllabus and Review	loc_31308C:		[ebp+var 4], eax	60 ; CODE XREF: sub_312FD8
				and	eax, OFFFFh eax, 80070000h	
				call	sub_3140F3	
			loc_31307D:			
					short loc_313080	
					short loc_313071 sub_3140F3	
					eax, eax	
					sub_3140F3	
			loc_31306D:			
					sub_31411B	
		, wargames/ba				
	http://overthewire.org	wargames/ha	ndfit ^{3/3066} :			; CODE XREF: sub 312FD8