Automatic Exp Generation	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D	
an Odyssey	esi eax, [ebp+arg_0] eax esi, 1D0h esi [ebp+arg_4] edi sub_314623 eax, eax short loc_31306D	
cmp jz loc_313066: push call		
امد عندی Sophia D'Antoine سی Hack.lu 2015	; sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C	
call and or loc_31308C: mov	sub_3140F3 eax, OFFFFh eax, 80070000h	CODE XREF: Sub_312FD8

Introduction

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 push eax push eax

Programs have become increasingly difficult

- larger, changing surface area
- mitigations
- more bytes to siphon through

ЦŲ	
st	
	short loc_31306D
	<pre>eax, [ebp+arg_0]</pre>
	[ebp+arg_4]
	sub_314623
	short loc_31306D
	[ebp+arg_0], esi
	short loc_31308F

			short 10c_31307D sub_3140F3 short 10c_31308C	
	loc_31307D:	call	sub 3140F3	
		and	eax, 0FFFFh eax, 80070000h	
0/22/2015	Program Analysis to Find Vulnerabilities			2/45 ; CODE XREF: sub 312FD8
			[ebp+var_4], ea	

Introduction

Reaction:

people get smarter and tools get better

- government research
- pentesters
- CTF!

		sub_314623		
		short loc 31306	D	
		[ebp+arg_0], eb:		
		short loc_31306		
		eax, [ebp+var 7		
		eax, [ebp+var_8		
		short loc 31306		
		eax, [ebp+var 8-		
	push	esi		
		[ebp+arg_0], eas		
		sub 31486A		
		short loc 31306	D	
a at h	ans 1			
		ear [ebp+arg 0]		
	push	eax		
		[ebp+arg 4]		
		sub 314623		
		short loc 31306	D	
		[ebp+arg_0], es:		
		short loc_31308		
Loc_313066:				
		sub_31411B		
Loc_31306D:				
		sub_3140F3		
		short loc_31307	D	
		sub_3140F3		
		short loc_31308	С	
Loc_31307D:				
		sub 3140F3		

Program Analysis to Find Vulnerabilities

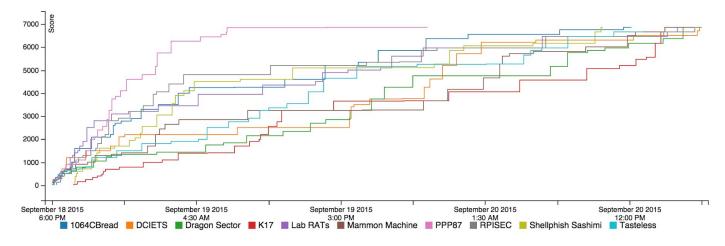
; CODE XREF: sub_312F.

lov [ebp+var

CTF & Wargames

SAW Home Rules Judges Teams Scoreboard Challenges Statistics CSAW.TV Archives

Scoreboard



All	Qualifying	High School	Industry	Undergraduate	Graduate	Other	
-----	------------	-------------	----------	---------------	----------	-------	--

Place	Team	Br	acket	Country	Score
1	1064CBread	Un	ndergraduate	United States	6860
2	DCIETS	Un	dergraduate Stacked	Canada	6860
3	RPISEC	Un	dergraduate Stacked	United States	6860
4	PPP1	Un	dergraduate	United States	5810
5	Shellphish Nigiri	Un	dergraduate Stacked	United States	5810

10/22/2015

Program Analysis to Find Vulnerabilities

Login

The Past

Manual labor

- static analysis
- dynamic analysis

dish edi all sub_314623 est eax, eax z short loc_31306D mp [ebp+arg_0], ebx nz short loc_313066 ov eax, [ebp+var_70] mp eax, [ebp+var_84] b short loc_313066 ub eax, [ebp+var_84] ush esi ush esi ush esi ush eax ush edi ov [ebp+arg_0], eax all sub_31486A est eax, eax z short loc_31306D ush esi ea eax, [ebp+arg_0] ush eax ov esi, 100h

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

CODE XREF: sub_312FD8 sub_312FD8+49

; CODE XREF: sub_312FD8

sub_3140F3 eax. OFFFFh

eax, 8007000

5/45

10/22/2015

Program Analysis to Find Vulnerabilities

; CODE AREF:

[ebp+var 4], eas

Dynamic Analysis

Definition

- Running it (concrete execution)
- Collecting/ observing environment changes

Popular Uses:

- dump VM memory & grep13066
- record/ replay & manual analysis
- gdb (debuggers) & run

		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]
		631
		[ebp+arg_0], eax
		sub_31486A
		short loc_31306D
•/		
		eax, [ebp+arg_0]
	push +	changes
IE	プロし	changes
		[ebp+arg_4]
		sub_314623
		short loc_31306D
		[ebp+arg_0], esi
		short loc_31308F
	push	ODh
וב	IICI	2110_31411B
	<u>ys</u>	Sub_31411B

10/22/2015

Program Analysis to Find Vulnerabilities

; CODE XREF: sub_312F

6/45

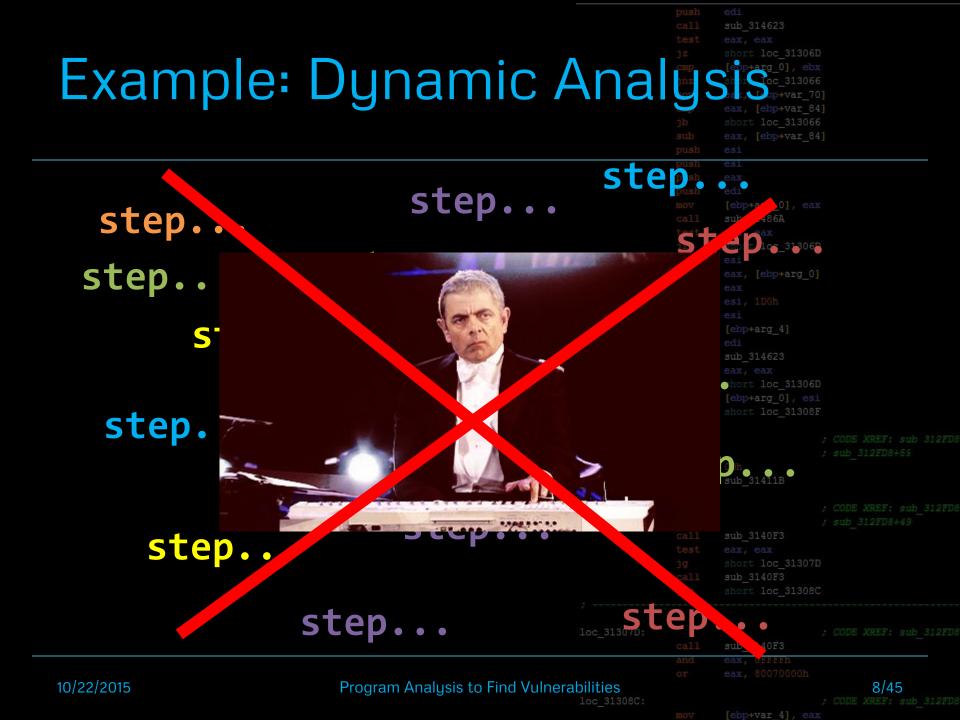
Dynamic Analysis

Common tools:

- gdb, windbg, cdb
- python brute force (blind fuzzing)

Program Analysis

			sub_314623	
			short loc_31306D	
			<pre>[ebp+arg_0], ebx</pre>	
SIS			short loc_313066	
			eax, [ebp+var_70]	
			eax, [ebp+var_84]	
			short loc_313066	
			eax, [ebp+var_84]	
		push	esi	
		push	esi	
			edi	
			[ebp+arg_0], eax	
			sub_31486A	
			eax, eax	
			short loc_31306D	
			esi	
			eax, [ebp+arg_0] eax	
A				
e (bline	d furz	zin	α	
	u iuz		Up+arg 4]	
			edi	
			sub 314623	
			short loc_31306D	
			[ebp+arg_0], esi	
			short loc_31308F	
	loc_313066:			
			sub_31411B	
	loc_31306D:			
			sub_3140F3	
			short loc_31307D	
			sub_3140F3	
			short loc_31308C	
	loc 31307D:			
			sub 3140F3	
		and	eax, OFFFFh	
to Find Vulneral	oilities			7/45
	loc_31308C:			
			[ebp+var_4], eax	



Automated Exploitation

Agenda

1.	Intro		p p
2.	Automating Exploitation		
	a. what, how?		
	b. the target		
С			
З.	Program Analysis		
	a. background		
	 b. types we care about 		
	c. how this helps with AEG	loc_313066:	
4.	Application		
	a. tools	loc_31306D:	
	b. demo		
F			
5.	Conclusion		
		loc_31307D:	c

es1 eax, OFFFFh eax, 80070000h 10/45

Automatic Exploit Generation

Some Background

What is Automated Exploitation?

The ability to generate a successful computer attack with reduced or entirely without human interaction.

- Focus on discovery and combination of write and read primitives
- Existing AE work focused on Restricted Models:
 - Sean Heelan's "Automatic Generation of Control Flow Hijacking Exploits for Software Vulnerabilities"

sub 314623

sub 3140F3

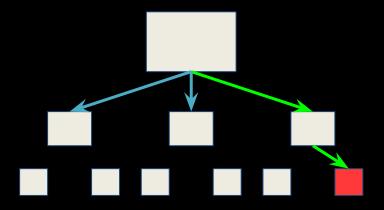
- David Brumley (@ Carnegie Mellon) et al. (AEG, MAYHEM, etc)
- Cyber Grand Challenge! (CGC)

	loc 31307D:				
			sub_3140F3		
		and	eax, OFFFFh		
10/22/2015	Program Analysis to Find Vulnerabilities			11/45	
	loc_31308C:				
			Lebration /1 es		

Automating Exploitation

Break up AEG into 2 parts:

- Generating **input** to **get to** vulnerability
- Generating "payload" to profit from vulnerability



		sub_314623	
		short loc_31306D	
atio		[ebp+arg_0], ebx	
Stin	n	short loc_313066	
allu	I IV	eax, [ebp+var_70]	
	ib	<pre>eax, [ebp+var_84] short loc_313066</pre>	
		eax, [ebp+var 84]	
	push	esi	
		[ebp+arg_0], eax	
		sub_31486A	
rability	jz	short loc_31306D	
rabiity	push		
	lea	eax, [ebp+arg_0]	
om vul	nera	bility	
	nush		
		[ebp+arg 4]	
		edi	
		sub 314623	
Both a	are	hard	
	jz	short loc_31306D	
Wonk	hoir	d d d d d	in
WOLK	DETI	ig done	
 223 (2003) 			
both a	area	IS	
		0.015	
Eacur	t_{0}	21/11/202	
Focus		lay on	
_loc 31306D:			
first	nrc	blem 🚽	
· -	call	sub_3140F3	
		short loc_31307D	
		sub_3140F3	
		short loc_31308C	
loc 31307D:			
		, sub 3140F3	
	and	eax, OFFFFh	
1 +1+1+			10/15

Program Analysis to Find Vulnerabilities

Automating Exploitation

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



10/22/2015

Automatic Exploit Generation

c, 80070000h

13/45

ov [ebp+var 4].

AEG - pwnable.kr

2)

4)

5)

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 push eax push edi mov [ebp+arg_0], eax call sub_31486A test eax, eax

sophia litterbox:~\$ nc pwn

; Attributes: bp-based frame sub_202795D proc near var C= byte ptr -0Ch

- Welcome to AEG (Autonat
- I will send you a newly co after you get the binary, when your input is given, you have 10 seconds to pui

ecx, al eax, cs:byte 222A323

eax, al esi, ecx

edi, eax sub 20278DD

retn sub_202795D end

locret_20279DD:

movzx

movzx

mov

eax, cs:byte_222A324

wait..

Get random binary, pwn it in 10 seconds.

1) Takes input at argv[1]

Program Operations

- Does some decode & operations on it
- 3) Calls sequence of 16 functions
 - Each function checks 3 characters of input sequentially
 - If you pass them all, you get to the exploitable memcpy!

Automated Exploit Generation

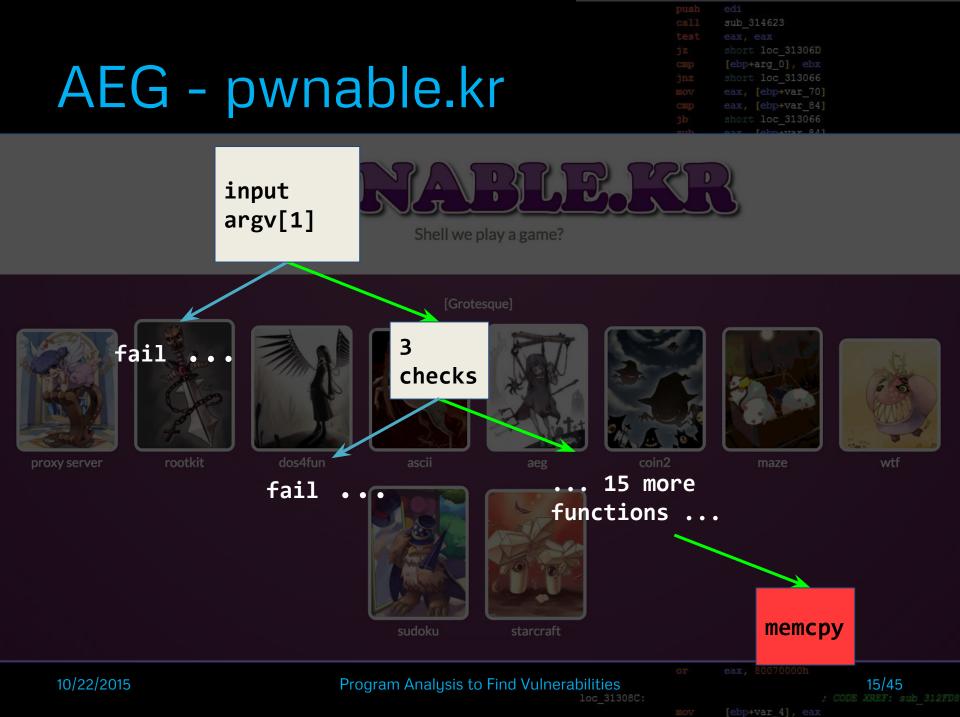
- 1) Generate input to get to vulnerability
- 2) Generate payload to exploit and get shell

10/22/2015

Program Analysis to Find Vulnerabilities

14/45

at



How can AEG solve for this path in the CFG?

Ø 3

Software Program Analysis!

Agenda

1.	Intro		p p
2.	Automating Exploitation		
	a. what, how?		
	b. the target		
3.	Program Analysis		
	a. background		
	b. types we care about		
		loc_313066:	
	c. how this helps with AEG		
4.	Application		
	a. tools	loc_31306D:	
	b. demo		
_			
5.	Conclusion		
		loc_31307D:	
			a

es1 eax, OFFFFh eax, 80070000h 18/45

Automatic Exploit Generation

; CODE A

What is program analysis

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

The process of **automatically** analyzing the behavior of applications

- In terms of a property:
 - program correctness

_	set of	naths ==	expected	naths
		putits	CAPCELEG	putito

program optimization

short loc_31306D
<pre>eax, [ebp+arg_0]</pre>
[ebp+arg_4]
sub_314623
short loc_31306D
<pre>[ebp+arg_0], esi</pre>
short loc_31308F

sub 31411B

sub 3140F3

; CODE XREF: sub 312FD8 ; sub_312FD8+59

; CODE XREF: sub_312FD0 ; sub_312FD8+49

minimum expense => expected paths

Program Analysis to Find Vulnerabilities

10	22	121	01	5
10		_		0

How This I	Helps with A	push call test jz curr curr curr jb sub push	short loc_3130 eax, [ebp+var_ esi	bx 66 70] 84] 66
automaticall - Fuzzing/ - Symboli	os us hunt for bu y. Instrumenting c Execution	push push push jz push lea push mov 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	6D 0] 6D Si
==> Pro mo	loc_31306D: OVE: COMbine a ; loc_31307D:	call call test Dice inconstruction call and	sub_3140F3 eax, 0FFFFh	; CODE XREF: sub_312FD8 ; sub_312FD8+49
10/22/2015	Program Analysis to Find Vulnerabilities		eax, 80070000h [ebp+var_4], e	20/45 ; CODE XREF: sub_312FD8

Types we care about.

Dynamic Binary Instrumen

Definition:

- 'Hijacked' environment, binaries, or urce
- Monitor specific system artifacts
- Attempts at complete (concrete) execution

Popular Uses:

- Force program states
- Gather and report observ
- Types of hooking: source & bin

		<pre>short loc_31306D [ebp+arg_0], esi short loc_31308F</pre>	
loc_313066:		; CODE XREF: sub 31 ; sub_312FD8+55 ODh sub_31411B	
ations ary		sub_3140F3 eax, eax short loc_31307D sub_3140F3 short loc_31308C	
loc_31307D:	call	; CODE XREF: sub_31 sub_3140F3	
arabilities		eax, 0FFFFh eax, 80070000h 22/45	

Program Analysis to Find Vulne

			sub_314623	
			short loc_31306	D
Exampl			[ebp+arg_0], eb	
Бхаллл			short loc_31306	
LAUNPI			<pre>eax, [ebp+var_7</pre>	
			eax, [ebp+var_8	
			short loc_31306	
			eax, [ebp+var_8	4]
		push	esi	
		push		
¢nin + inc	<pre>count0.so binary</pre>			
φρτιι -ι τιις	Counce.so Dinary		[ebp+arg_0], ea	
			sub_31486A	
			short loc_31306	
	— . .		esi	
BINARY LEV	ELI		eax, [ebp+arg_0	
		- Insh-		
- Inject	increment after each instru	ICT10	[ebp+arg_4]	
3			edi	
		call	sub 314623	
			short loc 31306	D
			[ebp+arg_0], es	
			short loc 31308	
STILL BRUTH	E FORCE			
-	loc_313066:			
Datura	total instructions for fur-			
- Keturn	total instructions for fuzz	equal	nput	
			sub_31411B	
– Only ti	rue for that 1 executed path			
	105 31306D:			
(the po	ossible CFG space may be ver	y la		
		Jg call	short loc_31307 sub 3140F3	
			sub_3140F5 short loc 31308	
			5110110 100_31300	
	loc_31307D:			
			sub 3140F3	
		and	eax, OFFFFh	
10/22/2015	Program Analysis to Find Vulnerabilities			23/45
	loc 31308C:			; CODE XREF: sub 312FD8
			Fobra unan (1) on	

Example: DBI

subu\$0xff, %edx
smþ %0xf f,%%døx
jłeunt++
mmp \$0x1, %edi
addu\$0x10, %eax
jle
icount++
mov \$0x1, %edi
icount++
add \$0x10, %eax

		sub_314623	
		short loc_31306E	
		[ebp+arg_0], ebx	
		short loc_313066	
		eax, [ebp+var_70	
		eax, [ebp+var_84	
		short loc_313066	
		eax, [ebp+var_84	
	push	esi	
		[ebp+arg_0], eax	
		sub_31486A	
		short loc_31306D	
		<pre>eax, [ebp+arg_0]</pre>	
		[ebp+arg_4]	
		sub_314623	
		short loc_31306D	
		[ebp+arg_0], esi	
		short loc 31308E	
6:			
		sub_31411B	
D:			
		sub_3140F3	
		short loc_31307D	
		sub_3140F3	
		short loc_313080	
		sub_3140F3	
	and	eax, OFFFFh	
			24/45
C:			

10/22/2015

Program Analysis to Find Vulnerabilities

[ebp+var 4], eax

Symbolic Execution

Definition:

- Generate 1 sym path for a set of paths (could still be extremely expensive)
- Satisfies path conditions
- Composed of some concrete values

Popular Uses:

- Determine program state at particular basic block
- Create 'equation' to feed to SAT/SMT solvers
- Faster than brute forcing all conditions

	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
	[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 312FD ; sub_312FD8+55

; CODE XREF: sub_312FD8 ; sub_312FD8+49

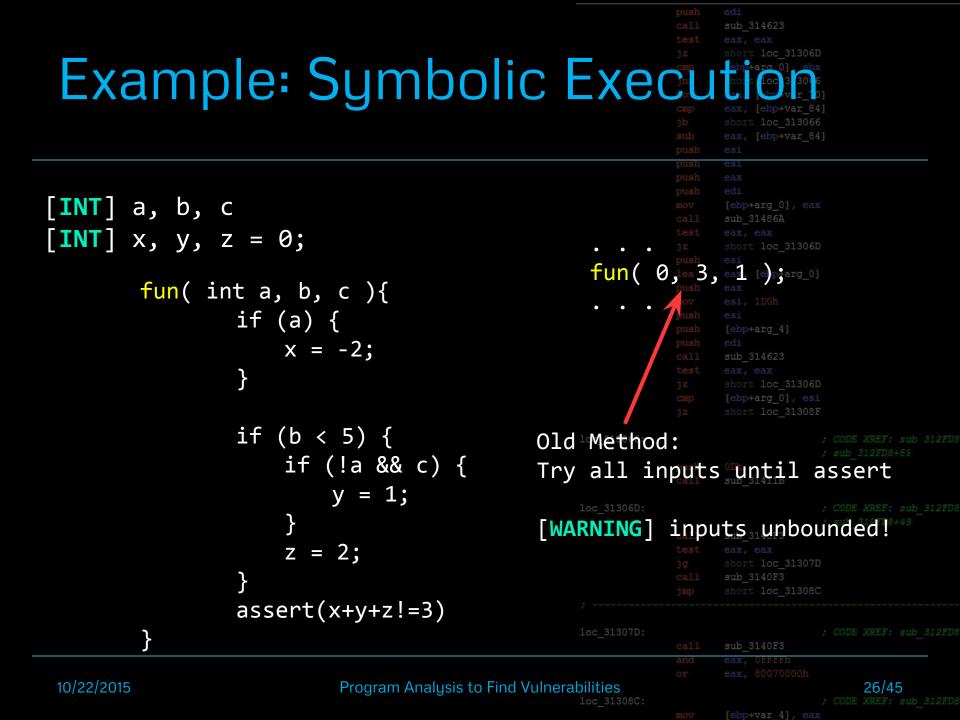
short loc_31307D sub_3140F3 short loc_31308C

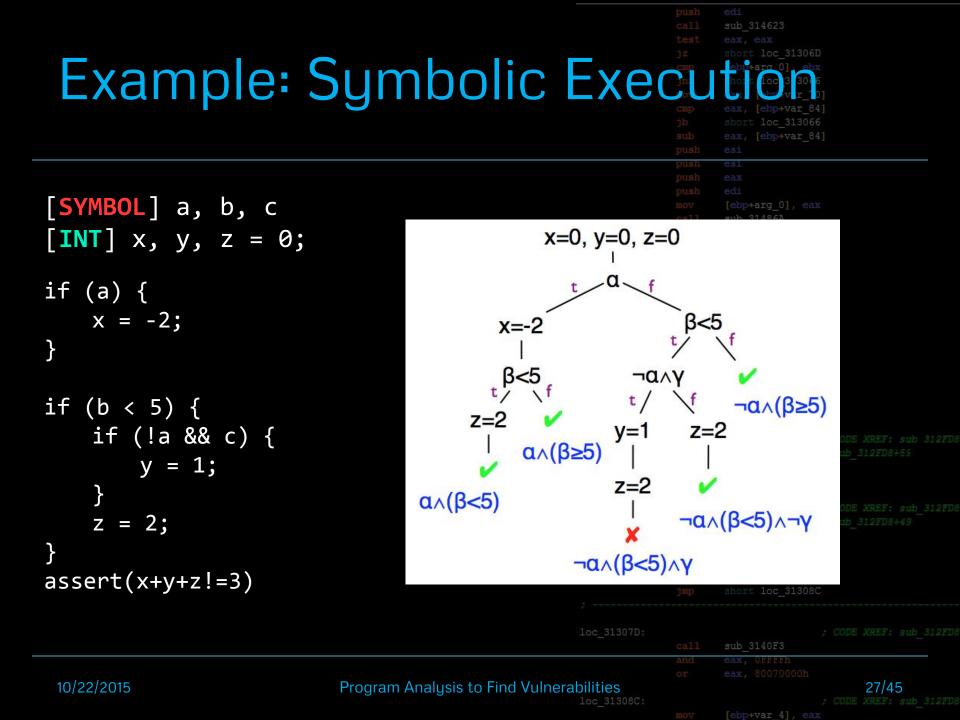
ax, OFFFFh

25/1

Program Analysis to Find Vulnerabilities

nov [ebp+var 4],





Concolic Execution

Definition:

- Dynamic symbolic execution
- Instrumentation of symbolic execution as it runs
- One path at a time to maintain concrete state underneath symbolic variables

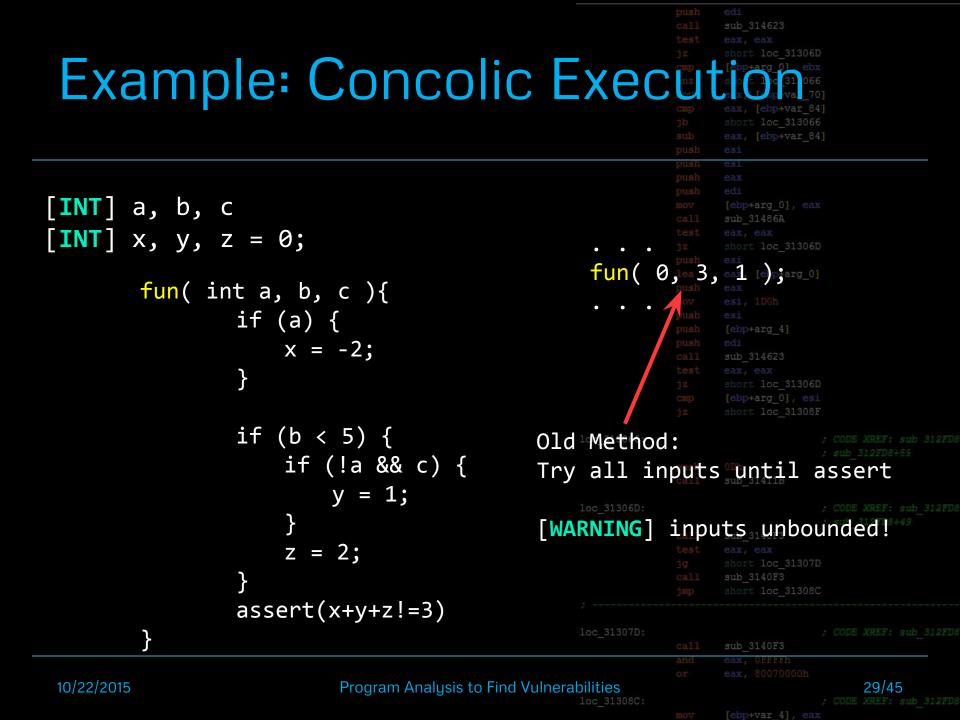
Popular Uses:

- Concretization (replace symbols with values to satisfy path condition)
- Handle system calls & library loading
- Cases which SMT can't solve

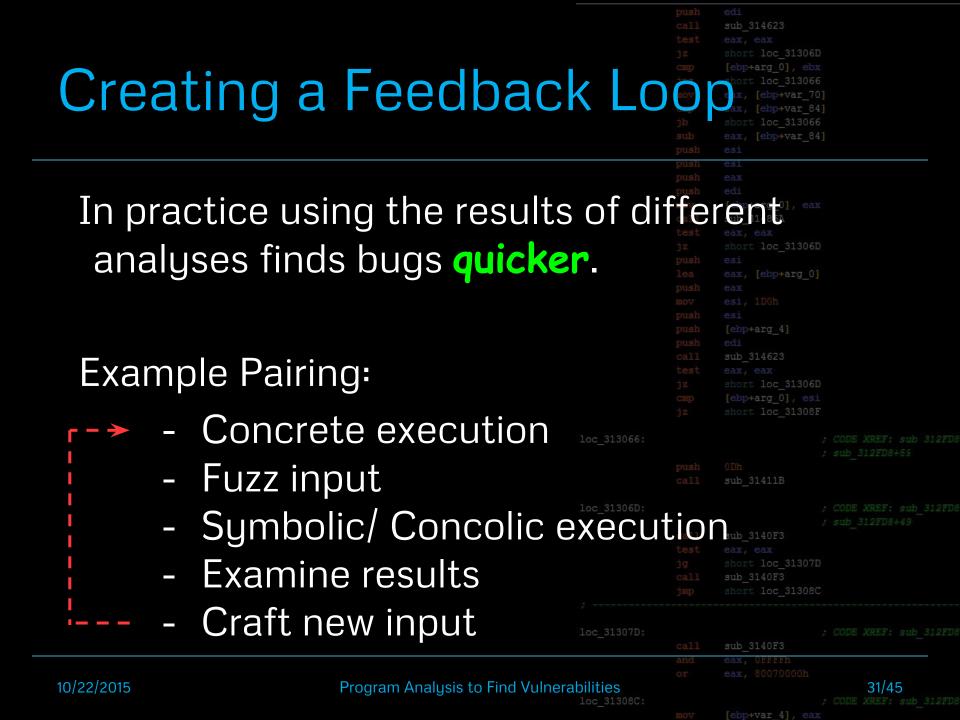
		sub_314623	
		short loc_31306D	
		<pre>[ebp+arg_0], ebx</pre>	
		short loc_313066	
		<pre>eax, [ebp+var_70]</pre>	
		eax, [ebp+var_84]	
		short loc_313066	
		eax, [ebp+var_84]	
	push	esi	
	push	es1	
		[ebp+arg_0], eax	
		sub_31486A	
		eax, eax	
		short loc_31306D	
1 •	lea	<pre>eax, [ebp+arg_0]</pre>	
ution	as I	truns	
1	push	esi	
ncret	e st	ate	
	push		
		sub_314623	
		short loc_31306D	
		[ebp+arg_0], esi	
		short loc_31308F	
oc 313066:			
		sub_31411B	
L: _ L			:
itisty	pat	n condit	ion) F: sub_312FD8
		sub 3140F3	
ling		eax, eax	
		short loc_31307D	
		sub 3140F3	
		short loc 31308C	
oc_31307D:		, sub 3140F3	
		eax, OFFFFh eax, 80070000h	
ilities			28/45
oc 31308C:			CODE VEET out 210ED4

Program Analysis to Find Vulnerabilities

ov [ebp+var 4], ea



Example: Co	oncolic Exec	push call test jz cmp inz inz jb sub push	<pre>short loc_313066 eax, [ebp+var_84] esi</pre>	
<pre>[INT & SYMBOL] a, b, c [INT] x, y, z = 0;</pre>	STEPS		esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi	
if (a) { x = -2;	concrete execution	of f	eax, [ebp+arg_0]	
<pre>} if (b < 5) {</pre>	[<mark>TWO</mark>] while building symb	push call collic	<pre>[ebp+arg_4] edi sub_314623 path_model [eop+arg_0], esi</pre>	
if (!a && c) { y = 1;	[THREE] constraints on inpu	jz ut ar	short loc_31308F ; CODE XREF: sub	
<pre>} z = 2; }</pre>	[FOUR] loc_31306D: models used to gene	erate	e concrete inpu	
assert(x+y+z!=3)	; loc_31307D:		short loc_31307D sub_3140F3 short loc_31308C ; CODE XREF: sub	
10/22/2015 Pr	ogram Analysis to Find Vulnerabilities	call and or mov	sub_3140F3 eax, 0FFFFh eax, 80070000h 30/45 ; CODE XREF: sub_ [ebp+var_4], eax	_



Agenda

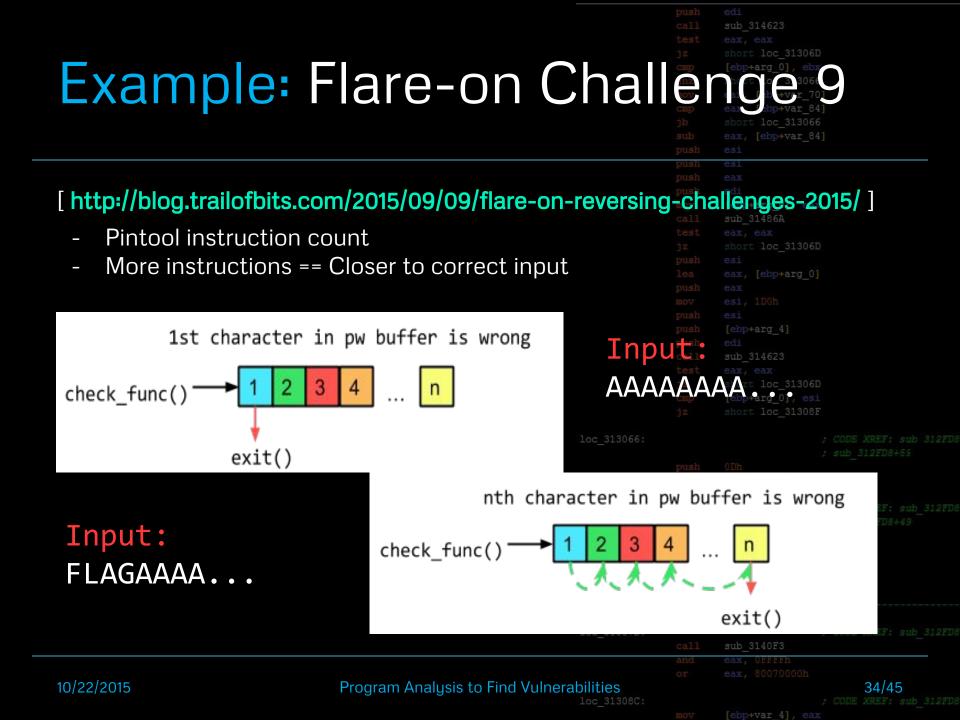
1.	Intro		E E
2.	Automating Exploitation		
	a. what, how?		
	b. the target		
3.	Program Analysis		
	a. background		
	 b. types we care about 		
	c. how this helps with AEG	loc_313066:	
4.	Application		
	a. tools	loc_31306D:	
	b. demo		
5.	Conclusion		
		loc_31307D:	. 6

eax, OFFFFh eax, 80070000h 32/45

Automatic Exploit Generation

es1

sub 314623 **Dynamic Binary Instrumentation** Common tools: PIN Tool Valgrind (before/during runtime) DynamoRIO Qemu 10/22/2015 Program Analysis to Find Vulnerabilities 33/45



Symbolic Execution

Common tools:

- KLEE (runs on LLVM bc)
- SAGE (MS internal tool)

Program Analysis to Find

feed it to z3 to solve

Vulnera	abilities			35/45 CODE XREF: sub_312FD8
		and	eax, OFFFFh	
			sub_3140F3	
	loc_31307D:			
			short loc_31308C	
			sub_3140F3	
			short loc_31307D	
			sub_3140F3	
	loc_31306D:			
			sub_31411B	
	100_010000.			
	loc 313066:			
			short loc_31308F	
			<pre>[ebp+arg_0], esi</pre>	
			short loc_31306D	
			sub_314623	
			[ebp+arg_4]	
			eax	
			eax, [ebp+arg_0]	
			esi	
			<pre>eax, eax short loc_31306D</pre>	
			sub_31486A	
			[ebp+arg_0], eax	
			edi	
		push	esi	
			eax, [ebp+var_84]	
			short loc_313066	
			eax, [ebp+var_84]	
			eax, [ebp+var_70]	
n			short loc_313066	
			[ebp+arg_0], ebx	
			short loc_31306D	
			eax, eax	
			sub 314623	

Concolic Execution

Common tools:

- Angr
- Pysymemu
- Triton

; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C				
<pre>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 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 lea eax, [ebp+arg_0] push eai call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5: / CODE XREF: sub_312FD8+55 push ODh call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C</pre>				
<pre>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 mov [ebp+arg_0], eax call sub_31486A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5: / CODE XREF; sub_312F ; sub_312FD8+55 push ODh call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 test eax, eax jg short loc_31308C</pre>			sub_314623	
<pre>cmp [ebp+arg_0], ebx jnz short loc_313066 mov eax, [ebp+var_70] emp eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi push esi 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 eax mov esi, 1D0h push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F ::</pre>				
<pre>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 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 eai push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			short loc_31306D	
<pre>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 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 eai push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			[ebp+arg_0], ebx	
<pre>cmp eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi 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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			short loc 313066	
<pre>cmp eax, [ebp+var_84] jb short loc_313066 sub eax, [ebp+var_84] push esi 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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			eax, [ebp+var 70]	
<pre>jb short loc_313066 sub eax, [ebp+var_84] push esi push esi push eax push edi mov [ebp+arg_0], eax call sub_31466A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5: : : : : : : : : : : : : : : : : : :</pre>				
<pre>sub eax, [ebp+var_84] push esi push esi 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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			short loc 313066	
<pre>push esi push esi 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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>push esi push eax push edi mov [ebp+arg_0], eax call sub_3146A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			esi	
<pre>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 eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>mov [ebp+arg_0], eax call sub_31486A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>call sub_31486A test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>			[ebp+arg 0], eax	
<pre>test eax, eax jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>jz short loc_31306D push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>push esi lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>lea eax, [ebp+arg_0] push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>push eax mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F s: : : : : : : : : : : : : : : : : : :</pre>				
<pre>mov esi, 1D0h push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>push esi push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F : : : : : : : : : : : : : : : : : : :</pre>				
<pre>push [ebp+arg_4] push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>push edi call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>call sub_314623 test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>test eax, eax jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>jz short loc_31306D cmp [ebp+arg_0], esi jz short loc_31308F 5:</pre>				
<pre>cmp [ebp+arg_0], esi jz short loc_31308F ; cODE XREF: sub 312F ; sub_312FD8+59 push ODh call sub_31411B call sub_31411B call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C ; cODE XREF: sub_312F call sub_3140F3</pre>				
<pre>jz short loc_31308F ;; ; CODE XREF: sub 312F ; sub_312FD8+59 push ODh call sub_31411B ;; CODE XREF: sub_312F ; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 ;; CODE XREF: sub_312F ;; CODE XREF; SUD ;</pre>				
5: ; CODE XREF: sub 312F ; sub_312FD9+55 push ODh call sub_31411B 0: ; CODE XREF: sub_312F ; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C 0: ; CODE XREF: sub_312F call sub_3140F3				
; sub_312FD8+59 push ODh call sub_31411B): ; CODE XREF: sub_312F ; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3): ; CODE XREF: sub_312F call sub_3140F3				
; sub_312FD8+59 push ODh call sub_31411B): ; CODE XREF: sub_312F ; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3): ; CODE XREF: sub_312F call sub_3140F3				
<pre>push 0Dh call sub_31411B):</pre>				
call sub_31411B): ; CODE XREF: sub_312F ; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C): ; CODE XREF: sub_312F call sub_3140F3				
<pre>call sub_3140F3 call sub_3140F3 call sub_3140F3 ig short loc_31307D call sub_3140F3 jmp short loc_31308C call sub_3140F3 call sub_3140F3</pre>				
; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C 0: ; CODE XREF: sub_312F call sub_3140F3				
; sub_312FD8+49 call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C 0: ; CODE XREF: sub_312F call sub_3140F3):			CODE XREF: sub 312FD8
<pre>call sub_3140F3 test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C 0: ; CODE XREF: sub_312F call sub_3140F3</pre>				
test eax, eax jg short loc_31307D call sub_3140F3 jmp short loc_31308C 		call		
jg short loc_31307D call sub_3140F3 jmp short loc_31308C 				
call sub_3140F3 jmp short loc_31308C): ; CODE XREF: sub_312F call sub_3140F3				
jmp short loc_31308C D: ; CODE XREF: sub_312F call sub_3140F3				
): ; CODE XREF: sub_312F call sub_3140F3				
call sub_3140F3				
call sub_3140F3				
and eax, OFFFFh		call	sub_3140F3	
		and	eax, OFFFFh	

Program Analysis to Find Vulnerabilities

; CODE XREF: sub_312

36/45

v [ebp+var 4],

AEG Demo: Assumptions

[Assumptions]

- Space of potential vulnerabilities too large
- Need to write tools to hunt for subset
 - Target memory corrupt (memcpy)
- ROP from there...

[Dynamically Acquire]

- Path to target
- Solve for constraints
- Addresses of gadgets for ROP

[Statically (Pre) Acquired]

- Semantics of target & gadgets

		sub_314623	
		short loc_31306D	
•	CIIID	[ebp+arg_0], ebx	
ptior		short loc_313066	
ULIUI	II W	<pre>eax, [ebp+var_70]</pre>	
	CIIID	<pre>eax, [ebp+var_84]</pre>	
		short loc_313066	
		<pre>eax, [ebp+var_84]</pre>	
	push	esi	
		[ebp+arg_0], eax	
		sub_31486A	
,		short loc_31306D	
		<pre>eax, [ebp+arg_0]</pre>	
		[ebp+arg_4]	
		sub_314623	
		short loc_31306D	
		<pre>[ebp+arg_0], esi</pre>	
		short loc_31308F	
loc 313066:			
		sub 31411B	
loc_31306D:			
		sub_3140F3	
		short loc_31307D	
		sub 3140F3	
		short loc_31308C	
loc_31307D:			
		sub 3140F3	
	and	eax, OFFFFh	
erabilities			37/45

Program Analysis to Find Vulnerabilities

LLVM Pass

Using the structure of the binary:

- **Dominator Tree**
 - Longest path of CFG is the "winning" path
- **Use-def chain**

Fast

Each cmp of this path comprises the "constraints"

\Rightarrow "Flow-sensitive constraint analysis"

LLVM:

10/22/2015

- Makes this analysis easier
 - DomTree & Use-def construction
 - **Semantics of cmp and vars** easy to pull out
 - Runs statically over bitcode (lift with Mcsema)

rg_0], eax 486A		
loc_31306D		
ebp+arg_0]		
rg_4]		
4623		
loc 31306D		
rg 0], esi		
loc_31308F		
411B		

- sub 3140F3

sub 3

sub 314623

loc 313066

eax,	01	t F	F.	11	



Program Analysis to Find Vulnerabilities

Angr Script

```
... acquire binary & some conditions ....
b = angr.Project("aeg")
```

```
ss = b.factory.blank_state(addr=entry_func)
ss.options.discard("LAZY_SOLVES")
ss.se._solver.timeout=10000
ss.memory.store(argv1_buff, ss.BV("input", 50*8))
```

```
pg = b.factory.path_group(ss, immutable=False)
angr.path_group.l.setLevel("DEBUG")
pg.explore(find=vuln addr[0], avoid=fail bbs)
```

100_313#setup env #fake input with no value

_313066:			
	sub_31411B		

#target & bad branches, 4 speed

sub 3140F3

argv1_win = pg.found[0].state.se.any_str(pg.found[0].state.memory.load(argv1_buff, 50))

#solved for path to target, dump memory

Program Analysis to Find Vulnerabilities

10/22/2015

; CODE XREF: sub___

39/45

Demo

Conclusion: The Future

[What We are (still) Working With]

- Binaries
- Source is nice
 - Need to lift bins to IR for LLVM
 - Most concolic exec. tools would need to compile it

sub 314623

[Difficulty]

10

- Know how to express our targeted vulnerability
- Semantics for UAF, Memory Corruption, etc....

			[ebp+var 4], ea	
22/2015	Program Analysis to Find Vulnerabilities			41/45 ; CODE XREF: sub_312FD8
		and	eax, OFFFFh	
			sub_3140F3	
	loc_31307D:			
			short loc_31308	C
			sub_3140F3	
			short loc_31307	D
			sub 3140F3	
	loc 31306D:			
			SUD_31411B	

Finding (More) Bugs

10/22

Automatic program analysis

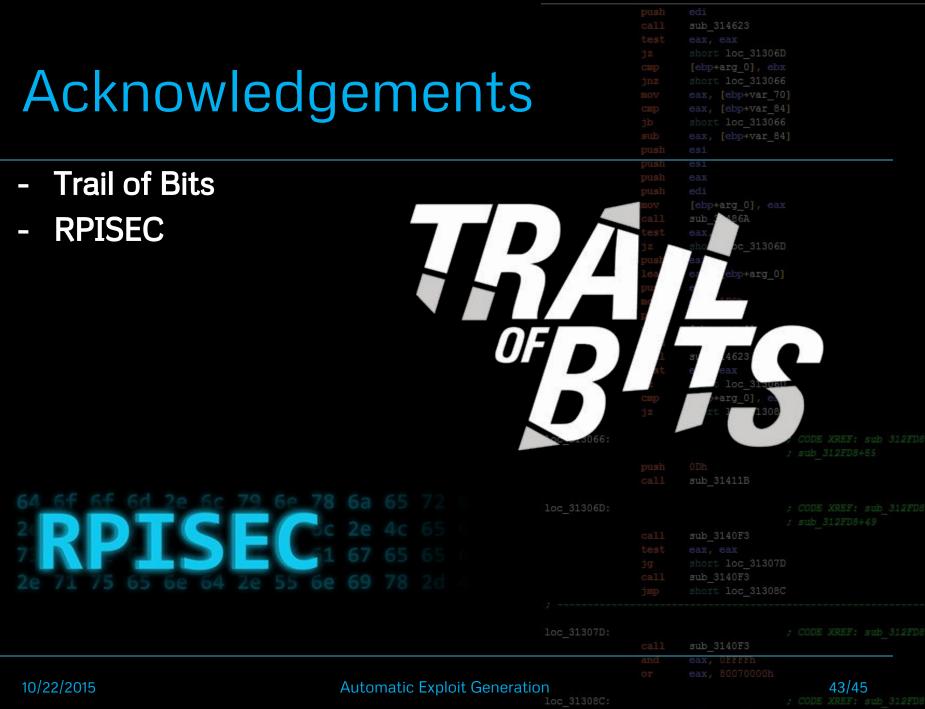
- translate program (IR)
- define program in-correctness

	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]
sh	esi
sh sh	esi
	esi eax
	esi eax
	esi eax edi
	esi eax edi [ebp+arg_0], eax
	esi eax edi [ebp+arg_0], eax sub_31486A
	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D
	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D
	esi eax edi [ebp+arg_0], eax sub_31486A eax, eax short loc_31306D esi

sub_314623
short loc_31306D
[ebp+arg_0], esi
short loc_31308F

goal: proving existence or absence of

2015	Program Analysis to Find Vulnerabilities			42/45 ; CODE XREF: sub_312FD8
			eax, OFFFFh eax, 80070000h	
		call	sub_3140F3	
	loc_31307D:			
			sub_3140F3 eax, eax short loc_31307I sub_3140F3 short loc_313080	
Nago	100_31308D:			



			sub_314623	
			short loc_31306D	
			[ebp+arg_0], ebx	
References			short loc_313066	
REIERINES			eax, [ebp+var_70]	
			eax, [ebp+var_84]	
			short loc_313066 eax, [ebp+var_84]	
			esi	
		push	es1	
[Good Course Mate	riall			
	гтат]		[ebp+arg 0], eax	
https://www.cs.umd.edu/class/	<u>spring2013/cmsc631/lectures/sv</u>	mholi	c-exec ndf	
https://www.utdallas.edu/~zxl?	<u>111930/spring2012/public/lec4.</u>	<u>pdt</u>	short loc_31306D	
http://web.mit.edu/16.399/www.	<u>/lecture 01-intro/Cousot MIT 2</u>	005 C	ourse 01 4-	1.pdf
<u>nttp://nomepage.cs.ulowa.edu/</u>	<pre>~tinelli/classes/seminar/Couso</pre>	<u>t.pat</u>	eax esi, 1D0h	
			[ebp+arg_4]	
			edi	
[cita fan Taal Da	aumontation]		sub 314623	
[Site for Tool Do	cumentation			
			short loc_31306D	
<pre>https://github.com/angr/angr-@</pre>			[ebp+arg_0], esi	
https://github.com/llvm-mirro	r/llvm		short loc_31308F	
	loc_313066:			
			0Dh	
[Other Good Resou	rces		sub_31411B	
	loc. 31306D:			
http://www.grammatech.com/blog	g/hybrid-concolic-execution-pa	rt-1		
	dia/people/jvanegue/files/aegc			
	uia/peopie/jvanegue/files/aegc	_vane	<u>gue.pur</u>	
			short loc_31307D	
			sub_3140F3	
			short loc_31308C	
	loc_31307D:			
		call	sub_3140F3	
			eax, OFFFFh eax, 80070000h	
10/22/2015	Automatic Exploit Generation		Gar, courocour	44/45
10/22/2013	Automatic Exploit Generation			44/45 CODE XREF: sub 312FD8
	100_010000.			

Any Questions?



IRC: quend email: sophia@trailofbits.com

10/22/2015

Automatic Exploit Generation

45/45