CTF Notes

Resource Sites:

* Hacking Resource: Null Bytes Hack like a Pro
* CTF Types: <https://www.endgame.com/blog/how-get-started-ctf>
* CTF Challenges<http://pwnable.kr/play.php>
* **Block of Memory:** <http://www.cprogramming.com/tutorial/virtual_memory_and_heaps.html>
* File Descriptor: integer num, created by kenerl to represent an opened file.
* Environmental variable: var, system information. Designaed var. w/ export. Ex: PWD, ~, etc.. . char\* envp[], argument C main function (Linux specific). Stores environmental var.
* Ssize\_t: filetype, represent size, blocks read/written, operation.
* Read(): C language, func, reads ≤ size bytes start from fldes. Store, result in buffer. Structure: ssize\_t read(int filesdes, void\*buffer, size\_t size). Return num. bytes read. size = 0 → read ∞ return 0.
* Buffer: memory, physical. Temporary storage mem. Between process. Usually RAM, quicker access.
* Hash function: func, takes input ∀ len. And output predefined len.
* Collision: occurrence, two inputs produce same hash output.
* Brute force: method, find answer, keep trying inputs.
* Collision Attack: hacking method, find two input string hash function, produce, same hash result. Collision, exploited, application compare two hashes. Generally, brute force, till collision. Ex: hacker, offer file download. Show hash validate download. Switch out file with other malicious file return same hash. Won’t obvious file validator diff. file.
* Int **p = (int**)num : p points to the member address at # num.
* \x##: python, converts hex# into a characture
* gets(): func read and store val in str point to by \*str, ln stdin till \n or eof
* flip-flop:
* Synchronous circuit: circuit, control, common clock line.
* Register (accumulator): group, flip-flops, store information. Synchronous circuit.
* process register: register, process, on microntroller.
* IP (instruction pointer): register, process. Place, computer at, program seq (Points, address next instruction).
* BP (Base pointer):
* buffer overflow(bof): overwrite, mem. frag. Of process. Overwrite IP or BP. Happen, buffer type char.
* Extended Stack Pointer: pointer,
* Extended Base Pointer: pointer,
* LIFO (last in first out): sequential execution, stack books.
* FIFO (first in first out): sequential execution, line.
* Stack: memory, static. Stored RAM. Var. alloc, stored memory. Access memory fast. Allocation, used compilation. All functions call each other and don't totlay execute last function return value. LIFO. Most recently reserved block first freed. Simple keep track stack. Free block adjust pointer.
* Heap: memory, dynamic. Stored RAM. Memory allocated, runtime. Acces memory slower. Heap only limited size virtual memory. Accessed randomly at any time. Allocate/ free block at any time. Complex track block.