Buffer Overflow Tutorial

<http://www.primalsecurity.net/0x0-exploit-tutorial-buffer-overflow-vanilla-eip-overwrite-2/>

<http://proactivedefender.blogspot.com/2013/05/understanding-buffer-overflows.html>

<https://itandsecuritystuffs.wordpress.com/2014/03/18/understanding-buffer-overflows-attacks-part-1/>

<https://medium.com/@johntroony/a-practical-overview-of-stack-based-buffer-overflow-7572eaaa4982>

https://zero-day.io/buffer-overflow-introduction/

EIP – Extended Instruction Pointer – Address of the Next Instruction

ESP – Extended Stack Pointer is the top of the stack

EBP – Extended Base Pointer is the bottom of the stack

1. Attach/Open Program to Immunity
   1. Hit Play
2. Crash Program
   1. Python Script
   2. python –c ‘print “A”\*1000’ | nc bof.local 9000
3. Verify Crash with EIP = 41414141
4. Create Pattern to Fill EIP With
   1. Pattern\_create.rb –l 1000
   2. !mona pc 1000
      1. Outputs to working folder C:\Users\Aidan Preston\Destkop\pattern.txt
5. Find Offset from EIP
   1. ./pattern.rb 31456791
   2. !mona pattern\_offset 0X7A46317A (Value of EIP at Crashtime)
   3. !mona findmsp
6. Control EIP
   1. If the offset is 230 for example this means we have to send 230bytes of data and then 4 bytes in our exploit, which will be a memory address of an instruction we want to execute
7. Find Existing JMP ESP Instruction to Use
   1. !mona jmp –r esp
   2. Set breakpoint at JMP ESP Instruction (maybe)
8. Find Bad Chars
   1. https://bulbsecurity.com/finding-bad-characters-with-immunity-debugger-and-mona-py/

Register - the register contains the contents of the different registers (e.g. EBX, ECX, EDX) and more importantly for our buffer overflow attack the **Instruction Pointer** **(EIP)** and the **Stack Pointer** **(ESP)**. The **Stack Pointer (ESP)** is where we will put our payload (shellcode), **the Instruction Pointer (EIP)** is where we put the address of the ESP (containing our shellcode), hence telling the program to execute our shellcode instead of doing what it would normally do.