# Hack `postgres` Source Code: Vol I

## Chapter 1 : C & Rust

### 1.40 Wild Pointer - C

When a pointer is declared in C, it points to some random memory location. And later, we initialize that declared pointer with some memory address that we know. As we initialized that pointer with the address that we know, we can handle that pointer with utmost control. But, if we leave that pointer at declaration stage itself where it is pointed to some random memory location, it is not possible to have more control. These type of pointers are labelled as `wild pointers` in C. People say that these kind of pointers makes a way for security vulnerabilities but none of them showed "how" in public platforms.. I guess we need to find it ourselves.

```
#include<stdio.h>

int main() {

    /*
    * Declaring variable.
    * Here, C assigns a randome memory address to p.
    */
    int *p;

    // Let's find the random memroy address.
    printf("%p", &p)

    /*
    * Let's find the data in random memory address by dereferencing.
    */
    printf("%d", *p);

   /* Let's assign some value to it.
   * If ranom memory location is not sensitive location,
   * There won't be any issue. But if it points to some
   * OS operations related memory address,
   * then there is possibility of wrecking part of OS.
   * I don't want to wreck the OS.. So..
   */

   //*p = "Commented to prevent crashing of my Laptop";

   return 0;

}
```