**Buffer Overflow Exploits**

A buffer overflow exploit is a situation where were using some probably low-level C function or something to write a string or some other variable into a piece of memory that is only a certain length.

It overwrites the memory addresses which cause some problems.

The operating system will call as a function, the main method of code. But actual process will be held in memory in a very specific way.

0xFFF is the equivalent of 11111111 memory address (32-bit address).

Here kernel will be command line parameters that we can pass to programme and environment variables.

**Memory Space**

Text is the one which is the actual code of programme

Data is the one where the variables are initialised and uninitialized.

Heap is the one where you allocate large things in memory.

Stack holds the local variables for each of functions.

Heap grows in the direction as you add memory and stack shrinks in opposite direction.

In stack, we have as follows:

* Cooling function
* Parameter
* Return address
* Base Pointer
* Buffer

Segmentation fault is occurred when we are accessing out of memory.