• **IPSEC**  
Point to point security scene  
Lives in network layer

• **Buffer Overflow**  
If I declare a variable with size of 6 bytes, and try to put 60 bytes.

• Basics:  
Programming error when a process attempts to store data beyond the limits of a fixed-sized buffer.  
Overwrites adjacent memory locations  
Buffer could be located on stack, heap or data section of the process.

• Consequences:  
Corruption  
Unexpected transfer of control  
Memory access violations  
Execution of code chosen by attacker

Still a major concern  
- So many code written in 70s, 80s.  
- Lazy, continued careless programming practices by programmers.

• Language:  
- C, C++ (memory management)  
  Has directly access to memory.

• Where:  
- Everywhere.  
Out of bound in c, segmentation fault.

Use Buffer Overflow as DoS attack.

• How to identify vulnerable programs:  
- Program source  
- Trace execution of programs as they process oversized input  
- Use tools

• Stack smashing  
Occur when buffer is located on stack  
- Return address  
- Local variables

If override the input, it will override the return address.

Target program can be:
A trusted system utility
Network service daemon
Commonly used library code

- Shellcode
  Overflow with code supplied by attacker.
  Attacker has to know the architecture.

- Defenses
  Compile-Time Defenses
  Run-Time Defenses

- Heap Overflow
  Attack buffer located in heap
  No return address

Avoid:
Memory management unit
Randomizing the allocation of memory on the heap
Critical Section (Guard pages)