Lab 15

Part A

Unauthorised data read and write

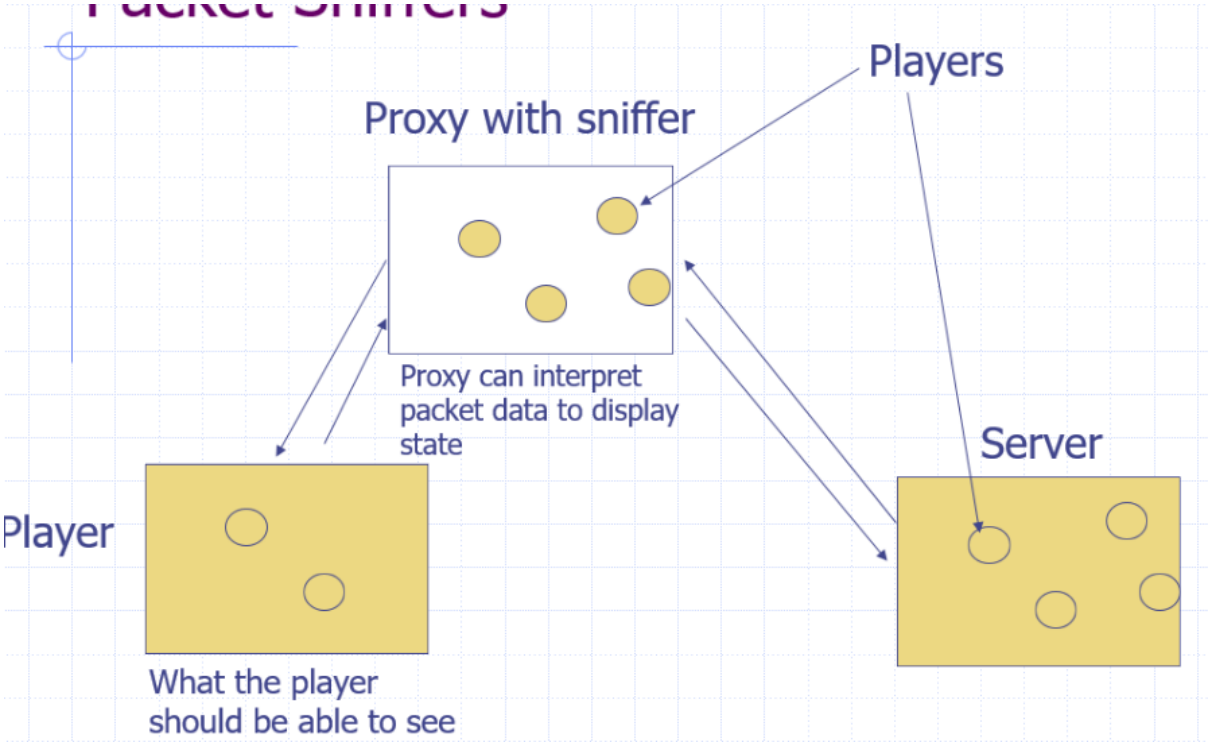
Packet Sniffers

In an online game, the clients typically receive more data than it needs at any one time.

- Client server - receive all data in our Area of interest (e.g. within the potentially hearable set)
- P2P - receive data from all other peers, even if not interested in particular peers

A packet sniffer can be used to read the data being received in each packet.

This can then be inerpreted by another program showing the state of the world such as using wireshark.



counteracting this hack

Packet level Modifications. Encryption of data transmitted along the wire.

There are two ways of detecting cheating on client side

- client side verification
- behaviour tracking

stack buffer overruns

- designed to foce the application to execute code injected by the attacker.
- when a function is called, the return address of the original calling function is saved to the stack.
- stack buffer overrun changes the return address to point to malicious code
- often malicious code preceded by block of NOP instructions

Counteracting the hack

typically, buffer overflow protection modifies the organisation of stack allocated data so it includes a canary value that, when destroyed by a stack buffer overflow, shows that a buffer preceding it in memory has been overflowed. by verifying the canary value, execution of the affected program can be terminated, preventing it from misbehaving or from allowing the attacker to take control over it.

Part B

From each location in the world, PVS is the set of regions that are potentially visible. In a static zone approach, only objects within the same static zones are considered relevant. However, in PVS approach neighbouring regions that are deemed potentially visible will contain relevant objects. The region sizes in PVS are typically mich smaller than separate static zones. A static zone might be a town or several buildings, while a PVS region would be an individual room inside of a building. The general goal of interest management is to reduce the cost of data communication in a distributed game. It improves the scalability of a game by limiting the amount of information transmitted to the players according to thir relevance.