**Last LAB**

1. In the context of IP restricted NAT devices, what does the hole-punching technique do? Describe this technique, using a diagram to aid your explanation if applicable.

Hole punching involves using a third-party server, referred to as a "rendezvous server," to help the two hosts establish a direct connection with each other.

Host A and Host B each send a request to the rendezvous server with their respective private IP address and port number.

The rendezvous server relays this information back to both hosts.

Host A and Host B then each send packets to the other's public IP address and port number, which creates a hole in the NATs.

If the holes are created at the same time, the NATs will allow the incoming packets to pass through to the private IP addresses of the hosts.

Host A and Host B can now communicate with each other directly.

A picture containing diagram, screenshot, line

Description automatically generated

1. In multiplayer online games, interest management cuts down on bandwidth usage by filtering relevant updates. Describe 2 common techniques of interest management.

1. Zone-Based Interest Management:

Zone-based interest management divides the game world into distinct zones or areas. Each player is assigned to a specific zone based on their location or proximity within the game world. Only updates relevant to the player's current zone are sent to them, minimizing the amount of data transmitted.

Here's how zone-based interest management works:

- Players receive updates about objects, events, and other players within their assigned zone.

- Updates outside their zone are filtered out, reducing unnecessary network traffic.

- As players move between zones, their assigned zone is updated accordingly, and they start receiving updates from their new zone.

Zone-based interest management is effective in open-world games where the game world is vast and players are typically interested in updates happening nearby. By focusing on relevant zones, bandwidth usage is optimized, allowing for more efficient data transmission.

2. Aura-Based Interest Management:

Aura-based interest management, also known as proximity-based interest management, considers the concept of an "aura" or a range around a player within which they are interested in receiving updates. Updates are sent to players based on their proximity to specific objects or other players within their aura range.

Here's how aura-based interest management works:

- Each player has an aura range defined around them, indicating the radius within which they are interested in updates.

- Updates are sent to players only for objects or players within their aura range.

- As players move, their aura range is updated, and updates are received or filtered based on the new proximity to objects or players.

Aura-based interest management is particularly useful in scenarios where players' interactions and interests are focused on nearby objects or other players. It helps reduce bandwidth usage by transmitting updates only for relevant entities within a player's aura range, ensuring a more efficient use of network resources.

Both zone-based and aura-based interest management techniques aim to optimize bandwidth usage by filtering out irrelevant updates. By transmitting only the necessary updates to players based on their location or proximity, these techniques enhance the multiplayer gaming experience while reducing the network overhead.

1. Interest management is important for good network performance on massively multiplayer games. What is a potentially visible set and how does this approach differ from static zones. How do these interest management approaches benefit the Game

A Potentially Visible set is a set of objects that are partially visible to the player based on the players position and current view.

Static zones divide the game into static areas that only allows updates for objects within the players current zone. In Potentially

Visible sets only the objects within the set are updated based on player movements and view not the zone the player is in.

These interest management approaches benefit the game in many different ways. Bandwith is freed up and the server does not need

to send as many updates as it would otherwise as the game only focuses on what is relevant. This allows the game to do more relevant things

and free up bandwith for more crucial updates

1. Describe one example for client-side attack and server-side attack, respectively. Please include details of how this attack works and a mechanism to prevent it.

Client-side = Phishing, the idea of phishing is where a fake or imposter website/link/email is created. Once the user clicks or accesses,

this fake site or link or email he would be prompted to insert sensitive data. The whole idea is to gain the users trust into thinking that

they are using the actual website owned by a legitimate company or an email from a legitimate source. Meanwhile it is all build by a scammer

Once the user enters any sensitive informations such as account credentials to bank or any other account, or even address or KYC system.

The fake website or resource captures all sensitive information and the creator has access to all this information. It is a common, scam

and infromation gain tactic.

To Prevent this users must know how to separate real from fake. How to identify legitimate websites, links and emails or messages.

There are many ways of doing this for example checking other parts of website functionality, very often the scammer is lazy and only builds

the login portion of a website, another way is analysing the link if it looks legitimate and checking the website certificate.

Ensure the connection is secure, another way of identifying issues is simply opening another window and googling the website to see if you

are using the real one or if the one the user is accessing seems illegitimate.

Server-Side = DoS Denial of Service. An attempt to make a service unavailable to its intended users. Overloading targets available

resources, disrupts network connection information. Two types of DoS attacks, Logic attack which exploits some software vulnerability and

Resource attack which overloads some resource suck as bandwith of the victim machine Using a VPN can prevent these as VPNs make a users real IP private while utilising

a different one.

1. Cheating in online games is the action of pretending to comply with the rules of the game, while secretly subverting them to gain an unfair advantage over an opponent.

Describe two ways of cheating. Please include details of how each cheating works and a mechanism to prevent it.

1 = Visual Assistance: Some variations = Wallhacks, Radarhacks

2 = Aim Assistance: Some variations = Silent Aim, AimBot, SpinBot, TriggerBot

For example AimBot is an extrenal software that a user downloads from a website and runs at the same time as the desired game.

For example the player wishes to cheat in Counter Strike: Global Offensive so the user downloads an AimBot created for the game.

The cheater runs the game and launches the cheating software also.

AimBot allows the user to gain an unfair advantage by perfecting the users aim . The user does not need to get good at a game,

the software simply aims at the enemies and all the player needs to do in this case is click shoot.

For a Visual Assistance example I will use wallhacks. Wallhacks is incredubly self explanatory. It simply shows the enemies to the

player through walls, objects and anything else within the game world.

It is incredibly powerful but as an observer very easy to spot when a player is looking at walls to where enemies are.

To prevent this game developers may opt to implement an anti-cheat into their game. Anti-Cheats vary greatly, some can be unintrusive but weak while others are incredibly intrusive

And run un kernel-level(Example: Valorant)

Some well known Anti-Cheats are VAC(Valve anti-cheat) and Easy Anti-Cheat

Anti-Cheats can work in different ways, the more intrusive ones check for what programs are being ran while a player is playing their game

to check for any cheating software, while less intrusive include analysing player performance and searching for any gameplay abnormalities

such as a player instantly killing all enemies, doing things that are impossible within the game or simple evaluationg a players game based on reports

or abnormal performance.