

# Captures d'écrans du code du système d'anti triche

Code pour bloquer les tricheurs qui se donnent des armes (dans le serveur de jeu, il est possible d'acheter des armes avec de l'argent réel. Les tricheurs cassent toute l'économie du serveur en faisant cela. J'ai donc réglé ce problème)

```
1 référence
private bool IsPlayerOwningWeapon(Player source, int weaponType, int weaponDamage, bool isNetTargetPos)
{
    if ((weaponDamage >= 5) && (isNetTargetPos))
    {
        uint hash = (uint)weaponType & 0xFFFFFFFF;
        if (WeaponsConfig.Weapons.TryGetValue(hash, out WeaponsConfig.WeaponInfo weaponInfo))
        {
            // On est obligé d'énumérer les armes car ox_inventory attend une string et non un hash
            // J'aurais pu refaire le système de JOAAT mais plus simple de les énumérer.
            int itemCount = GetItemCount(source.Handle, weaponInfo.Hash);
            if (itemCount > 0)
            {
                return true;
            }
        }
    }

    return false;
}
```

Patch d'une faille permettant aux tricheurs de faire quitter le jeu de tous les joueurs à proximité

```
warnings.cs  config.cs  wheels.cs  multipliers.cs  crash.cs  anticheat.cs  game.cs  anticheat.cs  deserialize.cs  noclip.cs  Server.csproj  Client.csproj  weaponconfig.cs  Class1.cs
Client
namespace Client
{
    0 références
    public class Wheels : Anticheat
    {
        1 référence
        public IEnumerable<Prop> GetNearbyObjects(Vector3 position, float radius)
        {
            var allProps : IEnumerable<Prop> = Game.GetGamePool<Prop>();

            return allProps
                .Where(prop =>
                {
                    float distanceSquared = prop.Position.DistanceToSquared(position);
                    float radiusSquared = radius * radius;
                    bool isInRange = distanceSquared <= radiusSquared;

                    return isInRange;
                }) // IEnumerable<Prop>
        }

        // Permet de flag quand un cheatteur utilise l'exploit avec les roues
        [Trig]
        0 références
        async Coroutine CheckWheels()
        {
            while (true)
            {
                if (Game.PlayerPed.IsInVehicle())
                {
                    Vehicle vehicle = Game.PlayerPed.CurrentVehicle;
                    if (vehicle.Owner.Character.Handle == Game.PlayerPed.Handle)
                    {
                        IEnumerable<Prop> objects = GetNearbyObjects(vehicle.Position, radius: 5.0f);
                        foreach (Prop obj in objects)
                        {
                            if (obj.Model == vehicle.Model)
                            {
                                vehicle.Delete();
                                Warnings.AddWarning();
                            }
                        }
                    }
                }

                await Wait(msecs: 0);
            }
        }
    }
}
```