AssaultCube HAX
Bryan Le, F16A-6841, 5361001, 6841

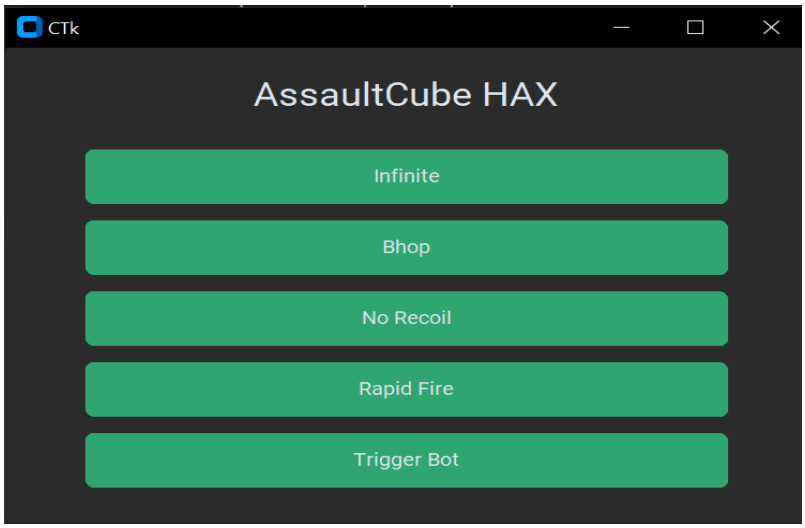
For my Something Awesome Project, I developed hacks for AssaultCube, a popular first-person shooter for beginner game hackers. Through the use of Cheat Engine, I was able to find the information I needed to write my scripts. The tool gave me the addresses and offsets of the values in the game such as a player's health, ammo, position etc. of which I could use to modify particular values however I wanted to.

Active	Description	Address	Type	Value
☐	Player			
☐	Player	P->00EBA090	4 Bytes	004E4A98
☐	Name	P->00EBA2B5	String[10]	unarmed
☐	Health	P->00EBA188	4 Bytes	100
☐	Force Attack	P->00EBA2B4	Byte	0
☐	Force Jump	P->00EBA0FB	Byte	0
☐	On Ground	P->00EBA0F9	Byte	1
☐	X-Y Position	P->00EBA098	Float	91
☐	Z Position	P->00EBA09C	Float	12.5
☐	X View Position	P->00EBA0D2	Byte	135
☐	Y View Position	P->00EBA0D4	Float	0
☐	Weapon			
☐	Rifle Rounds	P->00EBA1E0	4 Bytes	20
☐	Rifle Reserves	P->00EBA1B8	4 Bytes	40
☐	Rifle Firerate	P->00EBA208	4 Bytes	0
☐	Pistol Reserves	P->00EBA1A4	4 Bytes	50
☐	Pistol Rounds	P->00EBA1CC	4 Bytes	10
☐	Pistol Firerate	P->00EBA1F4	4 Bytes	0
☐	Grenades	P->00EBA1E8	4 Bytes	0
☐	Display Name	00501C38	String[20]	Shrike
☐	Pointer to Entity List	P->0D856968	4 Bytes	0
☐	Team	P->00EBA3BC	4 Bytes	1

With these addresses, I developed several scripts, including: infinite health and ammo, bunny hopping, no recoil, rapid fire and a triggerbot which was the most difficult to develop.

- The infinite health and ammo essentially gave me god mode and no matter how much I was shot, I could not die and I would never run out of bullets to shoot.
- Bunny hopping allowed me to hold down the space-bar and constantly jump, allowing me to increase my speed of travel with strafing.
- No recoil allowed me to shoot without having the bullets spray everywhere and keep the bullet spread in one place.
- Rapid fire increased the fire rate of my guns.
- Triggerbot allowed me to hover my crosshair over an enemy and begin shooting immediately.

Not only that, I also developed a mod menu GUI for the user of the cheats to more easily access and activate the cheats.



Results:
Initially, I knew absolutely nothing about game hacking. In this project, I was able to successfully use Cheat Engine to derive all my base addresses and offsets to be used later in the project as I realised early on that the addresses change every time the game is restarted. The cheat table above was my final cheat table after it was all done. After finding these addresses, I was able to develop the basic hacks, like the ones mentioned above in python rather than the usual c++. They required some basic reading and writing to memory to achieve. The most difficult hack by far was the triggerbot. It took much more complex logic and understanding the reading and writing memory to be able to achieve but eventually, I was successful. I then developed a GUI to make the cheats more accessible and allow them to run concurrently rather than as separate programs like the image above.

What I did:
Honestly, I spent my time horribly. I actually spent some time making CSGO hacks as that was my game choice before AssaultCube but CSGO got completely replaced with CS2 unfortunately. Though, I did make an early start with getting the addresses and learning how to use Cheat Engine but stopped working on it for a long time until the week before. My biggest issue was deciding whether to write it in c++ or python. I realised it was near-impossible for me to just learn c++ and then be able to write the cheats as well as creating DLL injections in time so I opted for python. It was still very difficult as there was almost no online resources for python in comparison to c++ and c# as they were the more popular choice but I was far more comfortable with python and learned how to make it work on my own through some basic understanding of c and c++ and being able to find alternatives for that in python.

How I was challenged:
I honestly learned so much and as stressful as this assignment was, I'm glad it was done. I learned how the world of game hacking was much more complex than I originally thought, even though I already believed it to be challenging. Coming into this, knowing where to start was the hardest part for me. I literally searched up "how to start game hacking" on Youtube to realise that I needed Cheat Engine as the first tool to start. And then I learned the purpose of Cheat Engine was to get the addresses inside the memory of the target games. After, I learned that cheats could be "injected" into the memory with DLL injections, which I knew absolutely nothing about and happened to stumble upon in a forum post from unknowncheats. At that time, I didn't even know the difference between external and internal hacks! But I understood the jargon and concepts better through research. Then I was at a crossroads between learning c++ or using python. Eventually, I learned that I could read and write memory with python despite there being very few resources for it. I had begun learning a bit of c++ during the project so when I saw some sample code on github, I was able to slightly understand it and adapt it to python. My main issue for my biggest problem was to just do consistent research on terms I didn't understand and watch as many Youtube tutorials and read forum posts because these eventually would lead me to the next step in game hacking. I learned that game hacking is a field that I enjoyed and I'm glad I chose this as my Something Awesome Project. My project management skills definitely need work - as it always has - as I need to work on it more consistently throughout rather than just start early and leaving it to the last minute. Next time, I'd like to take the time to learn the things I couldn't, including: reading and writing to memory in

c++, utilising dll injections, internal hacks with assembly code, writing more hacks and trying to game hack other games.