



HoGent

Faculteit Bedrijf en Organisatie

Anti-Cheating binnen Gaming & eSports voor FPS games die in lijn ligt met een online platform

Gianni Stubbe

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Jasper Dansercoer

Instelling: BeSports

Academiejaar: 2015-2016

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Anti-Cheating binnen Gaming & eSports voor FPS games die in lijn ligt met een online platform

Gianni Stubbe

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Jens Buysse
Co-promotor:
Jasper Dansercoer

Instelling: BeSports

Academiejaar: 2015-2016

Tweede examenperiode

Samenvatting

Binnen de sector eSports en gaming is cheating een van de grootste opkomende problemen. De gamewereld is een groeiende sector waarbij er meer en meer geld mee valt te verdienen waardoor er ook vaker wordt vals gespeeld. De spelers kunnen een onfair voordeel halen die heel miniem kan zijn tot spelbrekend. De bedoeling is dan ook om dit tegen te gaan en een competitie die gespeeld wordt te controleren op eerlijkheid via "anti-cheating".

In dit onderzoek gaan we op zoek naar de populairste methodes om vals te spelen: hoe deze werken, wat de verschillen zijn en een vergelijking van functionaliteit. Daarna nemen we een kijkje naar de eenvoudigste en beste manieren om dit tegen te gaan. Deze zullen getest worden aan de hand van meerdere cheats die er op los gelaten worden.

Als laatste onderdeel zou er ook graag een interview bij horen. Dit met een persoon die actief bezig is met de development van een anti-cheat tool. Dit om nog iets diepere informatie te werking van anti-cheats want het is niet eenvoudig om hier rond informatie te vinden omdat je zo ook valsspelers inzicht heeft in de anti-cheat.

Voorwoord

Ik wil BeSports bedanken voor het gebruik van de gameservers om testen op uit te voeren.

Inhoudsopgave

1	Inleiding	4
1.1	Probleemstelling en Onderzoeksvragen	5
2	Methodologie	6
3	Counter-Strike: Global Offensive	7
3.1	Algemeen doel van het spel	8
4	Cheats	10
4.1	Soorten	10
4.2	Wallhacks	10
4.2.1	Werking	11
4.3	ESP	12
4.4	Aimhacks	14
5	Anti-Cheats	16
5.1	Software	16
5.1.1	VAC: Valve Anti-Cheat	16
5.1.2	Sourcemod	17
5.1.3	SMAC: Sourcemod Anti-Cheat	17
5.1.4	Easy Anti-Cheat	17
5.1.5	Fairfight	17
5.2	Hardware	17
5.3	Sociale of Crowd-Sourced	18
6	De Test	21
7	Interview	22
8	Conclusie	23

Hoofdstuk 1

Inleiding

De bachelorproef speelt zich af binnen de eSports scene. eSports staat voor "electronic sports" (Dirs, 2015). Dit is een overkoepelende term voor alles dat met competitief gamen te maken heeft. Het kan vrij letterlijk gezien worden als een sport maar dan achter een computerscherm. Net als gewone professionele sporters hebben eSporters een strak trainingsschema die tot wel meer dan 12u per dag tijd in neemt. Wanneer er toernooien gespeeld worden valt er voor de spelers tot wel 18 miljoen dollar aan prijzengeld te winnen. (Valve, 2015)

Wat eSports ook zo kenmerkend maakt is dat het vrij nieuw is maar toch al heel veel mensen aantrekt. Het platform bij uitstek om eSports te volgen is "Twitch.tv", hiermee kunnen mensen matches, toernooien maar ook individuen bekijken die aan het gamen zijn. Op bepaalde momenten kunnen er tot wel enkele miljoenen mensen gelijktijdig kijken naar de grootste competities ter wereld op het vlak van gaming. Gemiddeld zijn er 500.000 mensen aan het kijken naar wat er zoal gespeeld wordt door andere mensen voor een totaal van 241.441.823.059 minuten over het jaar 2015. (Twitch, 2015)

Het genre game waar deze thesis zich op zal specialiseren zijn "FPS games". Dit staat voor First Person Shooters of in andere woorden, games waarbij mensen zich in een karakter wagen en kijken vanuit de ogen van deze persoon. De grootste game die momenteel competitief gespeeld wordt in de eSports scene (onder de categorie FPS Games) is de alom bekende Counter-Strike: Global Offensive. Met een 10 miljoen unieke spelers per maand heeft dit spel een heel grote spelersbasis en spelers van overal ter wereld. Dit is een van de meest bekeken games op Twitch en is ook een van de eenvoudigste games om te snappen vergeleken met de andere populaire eSports games. (Valve, 2016a)

Het nadeel aan deze game is het grote aantal valsspelers of "cheaters". Veel spe-

lers zijn snel geneigd om een van de veelvuldige cheats te downloaden en deze even te proberen wanneer ze een minder moment hebben. Het is dan ook helemaal niet moeilijk om deze te vinden na wat zoeken op het internet. Mensen downloaden deze dan even naar hen pc en schakelen deze dan in om voordelen te krijgen op het vlak van "aim" - het mikken met een geweer - of op vlak van "pressence" - mensen zien terwijl deze zich achter een muur bevinden (ook wel wallhack genoemd). Daarna gaat dit nog veel verder tot zelf de kleinste dingen zoals het rondspringen waardoor je meer snelheid haalt - bunnyhoppen genaamd -, dit is normaal gezien iets wat heel veel oefening bevat en zelf de beste spelers niet consistent kunnen doen.

Voor dit onderzoek zal er vooral geconcentreerd worden op het spel Counter-Strike omdat dit het meest gekend is en er dus ook het diepst op ingegaan kan worden. De technieken om vals te spelen en dit tegen te gaan zijn vrij gelijklopend tussen verschillende FPS games zoals Counter-Strike, Battlefield, Call of Duty,... De technieken die hier dan ook toegepast worden zullen parallel lopen voor deze voorgenoemde andere games.

Eerst wordt er ieder aspect uitgelegd van wat nodig is te weten om het spel te snappen. Daarna zullen we de veelvuldige valsspeel manieren en tools om dit tegen te gaan bekijken.

1.1 Probleemstelling en Onderzoeksvragen

Voor dit onderzoek gaan we dus op zoek naar een oplossing om cheaters tegen te gaan binnen FPS games toegepast op Counter-Strike. Dit met de bedoeling om op een online platform te implementeren zodat competities binnen games zo eerlijk mogelijk kunnen verlopen. Wat is de beste en meest praktische oplossing voor deze use case.

Waarom is het zo moeilijk om dit tegen te gaan en valt er iets te doen naast anti-cheating om dit tegen te gaan.

Hoofdstuk 2

Methodologie

In dit onderzoek ga ik eerst de lezer uitleggen wat de game Counter-Strike precies inhoudt. Het is moeilijk om ieder aspect van het onderzoek te snappen zonder dat de basis van het spel duidelijk is. Hierna wordt er eerst dieper ingegaan op de verschillende soorten cheats en wat meer over de werking hiervan. Vervolgens wordt er gekeken welke precieze manieren er zijn om dit tegen te gaan. Dit is opgedeeld in 2 hoofdcategorieën: eerst de softwaretools die vaak beter gekend zijn, daarna wat meer over een nieuwe oplossing met hardware.

Als laatste is het de bedoeling om een testopstelling te maken die op de servers test welke cheats gevangen worden door welke anti-cheats. Er wordt dan gekeken welke opties de beste oplossing biedt en of er een combinatie mogelijk is om zo veel mogelijk valsspelers tegen te gaan.

Hoofdstuk 3

Counter-Strike: Global Offensive

Voor we beginnen aan het onderzoek naar verschillende manieren om vals te spelen en om deze tegen te gaan moeten we eerst de basis van het spel snappen. In dit onderzoek gaat het om het spel Counter-Strike: Global Offensive of kort "CSGO". Deze game is een van de meest gespeelde competitieve games en staat er voor gekend om eenvoudig aan te leren zijn vergeleken met andere competitieve eSports games.

De voorganger van deze immens populaire game werd uitgebracht op 19 juni 1999. Origineel was dit een modificatie voor de uiteengezette en beruchte game "Half-Life". Veel games die vandaag de dag als alleenstaande verkocht worden zijn ooit begonnen als een speltype gemaakt door spelers die bouwden op de basis van Half-Life. Na deze mod heeft Valve (de huidige ontwikkelaar achter Counter-Strike en Half-Life) een samenwerking met deze spelers aangekondigd en kon de game als alleenstaand spel gespeeld worden.

Na vele beta's en heel wat patches werd de final versie gereleased. Deze release is een van de bekendste bij het competitieve onderdeel van Counter-Strike, deze staat gekend als "1.6". Dit werd snel een van de meest populaire competitieve computer games ooit en werd tot voor kort nog altijd gespeeld door veel professionele gamers hoewel deze toch al wat leeftijd had. (Te, 2014)

Hierna kwamen nog een andere versie genaamd "Counter-Strike: Source". Hoewel deze ook heel wat verkoop had is deze minder gespeeld geweest. De game waar wij ons op zullen concentreren is de in 2012 geïntroduceerde "Counter-Strike: Global Offensive". Deze versie van het spel is deze die vandaag de dag nog altijd gebruikt wordt.

3.1 Algemeen doel van het spel

Counter-Strike is niet zoals de meeste andere FPS games. Waar bij vele shooter games het vooral gaat om het gewoon neerschieten van de tegenstander en het aantal "kills" of "deaths" telt is dit bij Counter-Strike toch nog goed anders.

Bij Counter-Strike spelen er vijf spelers in twee teams, deze teams kunnen ofwel "Counter-Terrorist" (of CT) ofwel "Terrorist" (of T) zijn. Elk team heeft zijn eigen doel doorheen het spel. Heel eenvoudig kunnen we zeggen dat de Terrorists als doel hebben ofwel de ene bom die ze hebben tot ontploffing te brengen, ofwel door alle Counter-Terrorists te doden. De Counter-Terrorists daarentegen hebben dan als doel ofwel de bom onschadelijk te maken (als deze geplaatst werd) of om alle Terrorists te doden. Dit wordt herhaald tot een maximaal van 30 rondes die elk een 2 tal minuten duren. Aan 15 rondes (of halverwege het spel) worden de teams omgewisseld van kant, de Counter-Terrorists worden dan de Terrorist en omgekeerd. Het eerste team dat 16 rondes wint in totaal wordt de winnaar van de game gezien.

Een game wordt gespeeld op een bepaalde "map" of kaart. Deze bevat een locatie waar de Terrorist bij het begin van de ronde starten -Terrorists spawn- en een waar de CT's spawnen. Naast dit zijn er nog 2 locaties waar de bom geplaatst kan worden per kaart, dit zijnde: Bomb site A en Bomb site B. Daar kunnen ze binnen een bepaald afgebakend gebied de bom plaatsen om daarna de komst van de CT's af te wachten. Iedere map heeft ook unieke eigenschappen en maakt het daardoor vaak iets voordeliger om te starten aan een specifieke kant.

Dit klinkt nog allemaal niet zo moeilijk tot nu toe maar buiten deze regels heb je nog het geld die iedere speler heeft. Ieder team begint de eerste ronde met \$800 en kan daarmee kiezen om iets te kopen van wapen of iets van "utility" of "gear" zoals een "defuse kit" (hiermee kan wordt de tijd voor het onschadelijk maken van de bom gehalveerd) of een kevlar vest waardoor je minder schade krijgt. Omdat er in de eerste ronde niet veel geld ter beschikking staat kan er niet anders gekocht worden dan een pistool en wordt deze ronde ook wel de "pistol round" genoemd.

Iedere ronde krijgt een bepaalde speler geld voor zichzelf of voor het team aan de hand van wat er gedaan wordt. Voor een kill met een specifiek wapen komt er een bedrag bij de speler zijn pot. Hetzelfde voor het plaatsen van de bom. Maar als de bom tot ontploffing gebracht wordt krijgt het hele team geld. Het verliezende team krijgt dit dan ook maar een heel stuk minder.

Dit is een van de belangrijkste aspecten van het spel, het is heel handig als je kan inschatten of het ander team alles kan kopen die nodig is of als deze zullen moeten

sparen en dan de ronde erna alles zullen kopen. Het sparen en enkel spelen met het standaard pistool wordt ook wel "eco" genoemd. Als je als een van de laatste spelers overblijft in je team en het ander team heeft nog alle spelers in leven zal het wapen vaak gespaard worden voor de volgende ronde, dit noemt men ook wel eens "saven".

Hoofdstuk 4

Cheats

Om te snappen hoe we het valsspelen moeten tegengaan moeten we eerst begrijpen hoe het valsspelen precies werkt. Er wordt een korte introductie gemaakt bij iedere soort en wat ze precies doen. Verder wordt er uitgelegd waarom ze precies zo vervelend zijn.

De taal waarin vele cheats geschreven zijn is Assembly. Dit is een heel low level language. Dit gaat dus vaak inspelen op memory addresses en in direct contact met de CPU van een computer. Het nadeel aan deze taal is dat er vaak geen mogelijkheden die andere programmeertalen zo interessant maken zoals variabelen, overerving... (Swanson, 2001)

4.1 Soorten

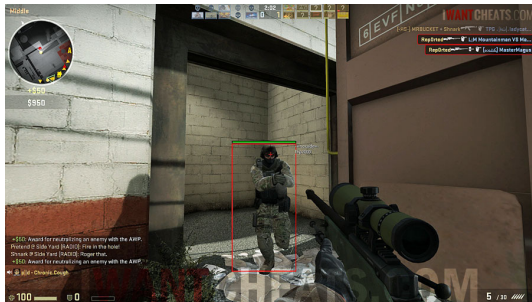
4.2 Wallhacks

De eerste onderscheiding die we maken zijn "wallhacks". Dit is een van de meest gebruikte termen onder spelers. Vaak wordt er mis melding gemaakt van deze cheats omdat ze het meest gekend zijn. Maar wat is het nu precies?

Als je je in het spel begeeft kun je met wallhacks heel eenvoudig iedereen zien lopen door de muur. Dit kan op verschillende manieren weergegeven worden. De eenvoudigste methode die door programmeurs gebruikt wordt is de box methode. Deze tekent rond ieder persoon een hokje die je door alle objecten in het spel kan zien. Zo kan je anticiperen waar de vijand zal verschijnen en heb je dus teveel oneerlijke informatie. De tweede manier die iets duidelijker te zien valt is met een outline rond de persoon die de vorm weergeeft hoe deze precies staat. Dit maakt het nog iets eenvoudiger voor de valsspeler omdat deze ook perfect weet waar de persoon zijn hoofd is, zijn armen enz.

Op die manier kan je dus heel eenvoudig al op iemand zijn hoofd gaan mikken. Ook kan de visibiliteit erg verbeteren waardoor je wederom het onfair voordeel haalt.

Hieronder even de 2 verschillen naast elkaar.



Figuur 4.1: Wallhack met een box (merk het rode hokje op rond de persoon)



Figuur 4.2: wallhack met outline

Een persoon met deze extra informatie kan op veel verschillende manier het spel beïnvloeden. Een van de eerste manieren om dit te doen is gewoon letterlijk volgen waar het andere team heengaat en anticiperen op deze situatie. Dit kan dan ook bijvoorbeeld megedeeld worden aan een teammaat die niet aan het valsspelen is. Zo kan je als enige cheater heel eenvoudig het volledige spel beïnvloeden.

4.2.1 Werking

Hoe is het precies mogelijk dat mensen weergegeven worden door de muur? Dat heeft te maken met de pakketten die tussen server en client worden verzonden. Iedere server verzend zijn pakketten naar de client, dit bevat de data van waar je teammates rondlopen, waar de vijand rondloopt, waar er smokes liggen en waar schoten geregistreerd zijn. De client zend zelf door wat er precies gebeurt achter de speler zijn pc. Dit zijn: waar die precies rondloopt, wanneer hij schiet en waar, welk wapen deze gebruikt en andere. Hoe de meeste gameservers werken is als volgt: alle data wordt doorgestuurd naar de client waar de onderdelen verwerkt worden. Dus de locatie van de niet zichtbare vijanden worden verzonden naar de client maar deze beslist wanneer deze weer te geven en wanneer niet.

Voor cheatontwikkelaars is het dus zo dat zij de data die de server verzend - maar normaal niet wordt weergegeven door de client - via hun software toch wordt weergegeven. Dit dan op de verschillende manieren zoals hierboven weergegeven.

Dit in Counter-Strike recent tegengegaan door de data die verzonden wordt door de server te minimaliseren. Enkel echt relevante data wordt doorgestuurd naar de client. Dus veel wallhacks werken hierdoor niet meer omdat niet alle data bij de gebruiker aanwezig is. De naam van de technologie hierachter is PVS of "Potentially Visible Set". Dit berekent wanneer een gebruiker een vijand kan zien en of horen en aan de hand daarvan wordt de data doorgestuurd. Op die manier is het enkel mogelijk voor cheats om personen weer te geven als deze ofwel door een teammaat gezien werden of wanneer je ze zelf ziet. Dit bestond al een heel stuk langer om de game te optimaliseren door te berekenen vanaf welke plaats je welke objecten kan zien maar nu wordt dit dus ook gedaan met de spelers zelf. (Valve, 2016c)

4.3 ESP

Als tweede cheat hebben we ESP wat staat voor "Extrasensory Perception". Extrasensory perception komt uit de wereld van de telepathie, wat gedefiniëerd kan worden als "the ability to know things (such as what another person is thinking or what will happen in the future) that cannot be known by normal use of the senses"(Merriam-Webster's Collegiate Dictionary, 2016). We vermelden deze als tweede omdat ESP heel vaak wordt gebruikt in combinatie met wallhacks. Net zoals bij de telepathie geeft deze extra hulp aan de gebruiker door informatie door te geven over de vijand. Dit niet over de locatie waar deze loopt maar vaak eerder over hoeveel geld de vijand beschikt en hoeveel health een bepaalde vijand nog heeft. Omdat dit heel 'droge' informatie is wordt dit vaak weergegeven naast een wallhack. Een voorbeeld kan je hieronder zien, let vooral duidelijk op het groene lijntje erbij. Dit is hoeveel levenspunten of "health" de vijand nog over heeft.



Figuur 4.3: Wallhack in combinatie met ESP (let op het groene balkje die aanduidt hoeveel leven nog resterend is per persoon)

4.4 Aimhacks

Wat het spel Counter-Strike nu zoveel anders maakt vergeleken met andere FPS-games is dat deze heel wat training vereist om ieder wapen te kunnen beheersen. Het is zo dat ieder wapen een bepaald uniek karakter heeft. Bij gewone geweren of "Rifles" is er vaak een patroon die deze volgt als je hem in 1 keer leegschiet, dit wordt het "spray pattern" genoemd. Dit is het omhoog gaan van het wapen in een bepaalde beweging wanneer je schiet. Omdat dit altijd hetzelfde is kan je ook dit tegengaan door je muis in de tegengestelde richting naar beneden en/of naar linksgs of rechts te bewegen. Een voorbeeld van zo'n spray pattern is hieronder afgebeeld voor een van de wapens.

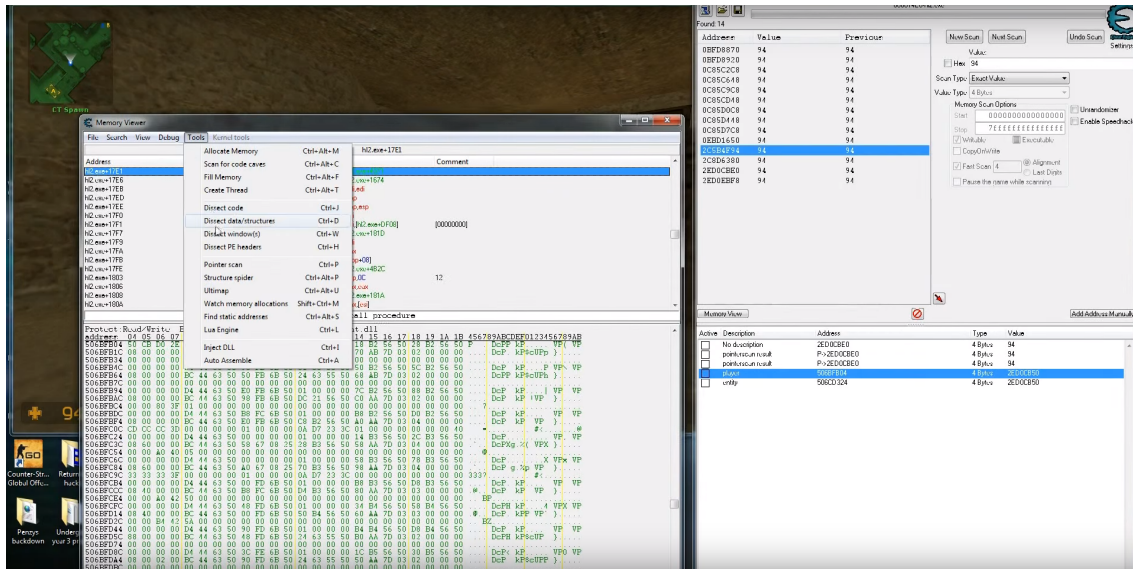


Figuur 4.4: Een spray pattern

Het is dus mogelijk om dit tegen te gaan door in omgekeerde richting dit tegen te gaan door de spray te controleren. Dit is een van de cheats die vaak bij aimhacks voorkomt. Zodra de gebruiker begint te schieten worden muisbewegingen gesimuleerd aan de hand van welk wapen ze precies vast hebben. Dan moet de gebruiker wel nog zelf de persoon zien te volgen natuurlijk. En dat is waar de echte aimhacks te pas komen.

De 2e vorm die vaker gebruikt wordt dan bovenstaande is iets moeilijker. Deze zal voor de speler mikken op een vijand en dan nog wel liefst op het hoofd. Hoewel deze moeilijker te maken is vanuit het programmeeraspect. Net zoals voorgaande

HOOFDSTUK 4. CHEATS



Figuur 4.5: Een voorbeeld van de gebruikte software om de geheugenadressering aan te spreken

cheats wordt ook dit in assembly geschreven. Een interessante reeks video's zijn deze van Guided Hacking (2013). Dit is een forum waarin de basis voor het maken wordt getoond voor het maken van een hack of cheat. In onderstaande afbeelding kan je duidelijk zien dat er gewerkt wordt met de geheugen adressen van het systeem.

Hoofdstuk 5

Anti-Cheats

Het belangrijkste aspect van deze bachelorproef is het tegengaan van het valsspelen. Dit is waar we nu op in zullen gaan. Dit onderdeel wordt opgedeeld in 3 hoofdcategorieën. Eerst de software tools die op de gameserver draaien. Ten tweede heb je dan ook iets nieuw en opkomend: de anti-cheat hardware. Als laatste hebben we nog de sociale of crowd-sourced anti-cheats.

5.1 Software

Als eerste zullen we even de software overlopen. Dit is de meest voorkomende oplossing en heeft dus ook een waaier aan oplossingen. Omdat het belangrijk is dat anti-cheat software mee is met de laatste vernieuwingen is er hoofdzakelijk geconcentreerd op software die nog altijd actief in development is of lijkt.

5.1.1 VAC: Valve Anti-Cheat

De grootste en meest bekende software is de Valve Anti-Cheat software. Dit komt standaard bij iedere installatie van de game en ook iedere server runt dit standaard wanneer deze opgestart wordt. (Valve, 2016d) Omdat iedere speler deze software staan heeft is dit ook de software die de meeste gebruikers pakt absoluut. In totaal zijn bijna een vier miljoen bans uitgedeeld. (xPaw and Marlamin, 2016)

Hoewel dit heel doeltreffend is kan er toch nog veel beter gewerkt worden aan deze software. In 2014 kwam er een schandaal naar buiten waarbij 3 professionele spelers langere tijd gecheat hadden zonder dat dit gemerkt werd door het Valve Anti-Cheat systeem. Tot een plots moment dan toch, daarna werden deze plots gevangen door VAC en was er heel veel ophef hier rond. Daarna was de vraag die iedereen stelde: wie is de volgende. Er werd een heksenjacht georganiseerd op alle andere professio-

nele spelers. Er was lange tijd twijfel of iedereen wel volledig zelf en eerlijk speelden. Hoewel de gemoederen gestild zijn is er toch vaak nog sprake over twijfel die onder de kijkers speelt. (Lahti, 2014)

Dus als serveradministrator wordt er vaak gekozen voor een third party tool die dit moet tegengaan. Deze zijn vaak nog iets beter en vaker geüpdate.

5.1.2 Sourcemod

Een van de belangrijkste tools die nodig is om geavanceerde wijzigingen door te voeren op een gameserver met de Source engine (de game engine van Valve) zijn plugins meer bepaald Sourcemod. Deze biedt nog veel meer functionaliteit aan dan anders mogelijk zou zijn. Er zijn dan ook enkele software pakketten die deze gebruiken om hun functionaliteit doeltreffender te maken.

Sourcemod is heel eenvoudig zelf te installeren. Men download de benodigde files plaatst deze via FTP op de gameserver in de juiste map en herstart de gameserver. Nadien kun je via commando's nagaan of dit correct werkt. Eenmaal dit correct is ingesteld is het eenvoudig om anti-cheats toe te voegen en of verwijderen.

5.1.3 SMAC: Sourcemod Anti-Cheat

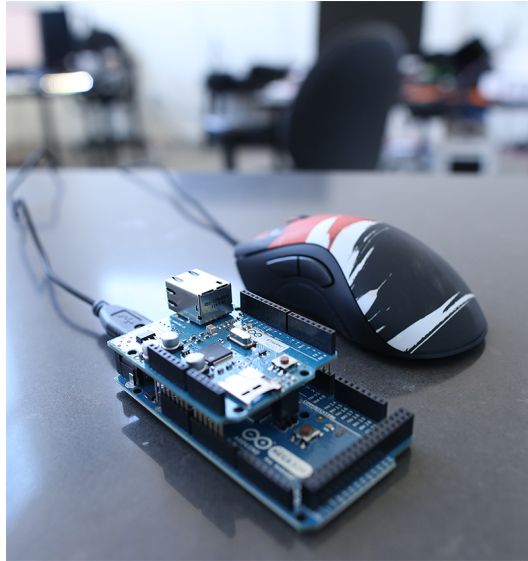
5.1.4 Easy Anti-Cheat

5.1.5 Fairfight

5.2 Hardware

Iets wat sinds 2014 aangekondigd is, is de Game:ref. Dit is een klein doosje die binnenin een arduino bordje bevat. Wat doet dit juist? Je sluit dit apparaatje aan op je computer en op het internet. Daarna sluit je op het apparaatje je muis aan. Deze heeft het signaal van de muis door aan de computer dus een speler zal er niets van merken. Het apparaatje vergelijkt dan via de gameserver (daarom de internetconnectie) of wat de persoon input via zijn muis wel gelijk is aan wat de server doorkreeg. Als dit niet zo is kunnen we concluderen dat er iets gebeurt tussen de muis en de gameserver wat dus duidt op een cheat.

Het concept zit heel goed in elkaar en zou specifiek aimhacks tegengaan (wallhacks kun je hier niet mee tegenhouden). Het enige probleem hiermee is dat niet iedereen zoiets zal aankopen, laat staan als deze wel zou cheaten, dan ga je echt jezelf niet



Figuur 5.1: De Game:ref

aangeven. Het lijkt dan ook meer een product gericht op evenorganisators of organisaties.

De campagne ging van start vorig jaar in april op Kickstarter maar is vrij snel gefaald en weer offline gehaald. Uit een post van de maker op de facebook pagina blijkt wel nog dat dit product in ontwikkeling is dus het kan in de toekomst zeker een interessant product zijn om mensen te controleren die meedoen aan een evenement zoals de professionele spelers. (TITARENCO, 2014)

Hoewel hier niet mee getest kan worden is dit zeker iets dat wel in overweging moet genomen worden voor toekomstige onderzoeken. Het is iets volledig nieuw en als het concept werkt kan het veelbelovende resultaten halen.

5.3 Sociale of Crowd-Sourced

Een van de laatste manieren om aan anti-cheating te doen is via Sociale anti-cheats of ook wel "Crowd-Sourced". Dit zijn systemen die gebruikers laat kijken naar verdachte personen en hen laat beslissen of deze al dan niet valsspelen.

Hiermee zijn wel 3 problemen:

1. Eerst en vooral: hoe beslis je wie verdacht is en wie niet. Controleer je iedereen of laat je enkel aangeduide spelers gecontroleerd worden.

2. Ten tweede: wie laat je deze bekijken? Kan iedereen dit of zijn er toch vereisten zodat niet jan en alleman iemand kan veroordelen tot valsspelen. Want wat we ten zeertste willen vermijden zijn mensen die onrechtvaardig verbannen worden.
3. Als derde moeten we ook rekening houden met wanneer je beslist iemand te verbannen van je gameservers. Doen we dit wanneer er een iemand zegt dat deze valsspeelt op basis van 1 kijker? Of laten we meerdere mensen kijken en kijken we hoeveel procent van deze zegt dat de verdachte valsspeelt?
4. En als laatste: Hoe maak je dat mensen dit blijven doen en dit geen vergeten functie wordt? Geef je er iets voor in ruil of hoe houdt je mensen aangetrokken tot de functie.

Dit wordt standaard wel al gedaan bij de laatste versie van Counter-Strike. Deze optie noemt "Overwatch" in het spel. Hoe zij dit beschrijven is als volgt: "The Overwatch lets the CS:GO community regulate itself by allowing qualified and experienced members of the community ('investigators') to review reports of disruptive behavior, determine whether those reports are valid, and apply temporary bans if appropriate." (Valve, 2016b)

Hoe werkt dit nu precies? Mensen van een bepaalde rank binnen het spel en die een bepaald aantal spellen al gespeeld hebben krijgen toegang tot de functie. Eenmaal je dit hebt komt er in het start menu een knop waar je een "Case kan reviewen". Dit download dan een video bestand waar alle namen van spelers zijn uitgewist. De verdachte noemt "The Suspect" en je krijgt een videofragment te zien van een 10 tal rondes. Hierna sluit de video af en krijg je per cheat categorie de optie om te kiezen of jij denkt of deze valsspeelt of niet. Deze case komt ook bij andere spelers terecht, wanneer een bepaald aantal bevestigd (het is onbekend hoeveel precies of procentueel) dan wordt "The Suspect" verbannen. Ze proberen je dit te doen aanhouden door je een melding te geven wanneer er iemand verbannen wordt die jij veroordeeld hebt. Op deze manier weten spelers dat ze de game zelf zo van cheaters proper houden en moet dit hen de neiging geven om dit vaker te doen.

De beslissing wie verdacht is wordt genomen aan de hand van de spelers zelf. Als je een officiële match speelt kan je iemand raporteren voor verdacht gedrag. Wanneer een bepaald aantal spelers dit doet komt deze terecht tussen de Overwatch cases.

Overwatch Resolution

Now that you've reviewed the evidence, it is your duty to determine the facts for each of the categories below. Note that for each category of disruption you should **presume the suspect to be innocent**. Only if you are confident that you witnessed behavior that would be agreed upon by the CS:GO community to be disruptive, anti-competitive, and/or anti-social **beyond a reasonable doubt** should you select **Evident Beyond a Reasonable Doubt**. If you are not 100% confident in your decision, you should select **Insufficient Evidence**.

If you don't feel that you were able to dedicate the appropriate attention level required to reach a verdict, please click on **POSTPONE JUDGEMENT**. The case will remain available for you to review at a later time.

Major Disruption: Aim Assistance

The evidence you reviewed would be judged, by any reasonable member of the CS:GO community, as clearly demonstrating that The Subject **used external software to improve their aim**, e.g. automated targeting, recoil reduction or elimination, etc.

☐ Insufficient Evidence
 ☒ Evident Beyond a Reasonable Doubt

Major Disruption: Vision Assistance

The evidence you reviewed would be judged, by any reasonable member of the CS:GO community, as clearly demonstrating that The Subject **used external software to gain information about locations of their opponents**, e.g. vision through walls or smoke, flashbang effect reduction or elimination, etc.

☐ Insufficient Evidence
 ☒ Evident Beyond a Reasonable Doubt

Major Disruption: Other External Assistance

The evidence you reviewed would be judged, by any reasonable member of the CS:GO community, as clearly demonstrating that The Subject **used external software other than those listed above to gain an advantage over their opponents**, e.g. speedhacking, automated jumping scripts, upside-down views, etc.

☒ Insufficient Evidence
 ☐ Evident Beyond a Reasonable Doubt

Minor Disruption: Griefing

The evidence you reviewed would be judged, by any reasonable member of the CS:GO community, as clearly demonstrating that The Subject **exhibited behavior that was disruptive, anti-competitive, and/or anti-social**, e.g. deliberately interfering with or trying to damage or be damaged by teammates.

☐ Insufficient Evidence
 ☒ Evident Beyond a Reasonable Doubt

Figuur 5.2: Een voorbeeld van de opties na het bekijken van een Overwatch video

Hoofdstuk 6

De Test

WIP

Hoofdstuk 7

Interview

WIP

Hoofdstuk 8

Conclusie

Bibliografie

Dirs, B. (2015). Is computer gaming really sport?

Guided Hacking, . (2013). How to make an aimbot c++ pt 1/6 tutorial.

Lahti, E. (2014). Cs:go competitive scene in hacking scandal, 3 players banned.

Merriam-Webster's Collegiate Dictionary, . (2016). Merriam webster dictionary: Extrasensory perception.

Swanson, W. (2001). Introduction to assembly language.

Te, Z. (2014). Dust to dust: The history of counter-strike.

TITARENCO, D. (2014). The world's first hardware anti-cheat device.

Twitch (2015). Twitch retrospective 2015.

Valve (2015). The international.

Valve (2016a). Counter-strike: Global offensive blog.

Valve (2016b). Overwatch faq.

Valve (2016c). Pvs.

Valve (2016d). Valve anti-cheat system (vac).

xPaw and Marlamin (2016). Latest game and vac bans.

Lijst van figuren

4.1	Wallhack met een box (merk het rode hokje op rond de persoon) . . .	11
4.2	wallhack met outline	11
4.3	Wallhack in combinatie met ESP (let op het groene balkje die aangeeft hoe veel leven nog resterend is per persoon)	13
4.4	Een spray pattern	14
4.5	Een voorbeeld van de gebruikte software om de geheugenadressering aan te spreken	15
5.1	De Game:ref	18
5.2	Een voorbeeld van de opties na het bekijken van een Overwatch video .	20

Lijst van tabellen