Undersökning av Rainbow Six Siege i WireShark

David Albertsson, da4768al-s@student.lu.se
Christer Andersson, ada07can@student.lu.se

Abstract— **Denna undersökning tittar på trafiken som spelet Rainbow Six Siege skickar respektive tar emot och analyserar mängden data samt vilka protokolltyper som används i olika situationer. Med hjälp av WireShark analyserades nätverkskommunikationen och aktuell data filtrerades ut från den ovidkommande trafiken för ett optimalt resultat. Den insamlade datan sparades undan för att möjliggöra en djupare analys och analyserades sedan. Undersökningen visade att vid single player skickades som förväntat mindre data än vid multiplayer. UDP var det primära protokollet för kommunikation men TCP och RTP kunde även observeras i mindre mängder. Slutligen så fann undersökningen att Rainbow Six Siege effektivt väljer det protokoll som passar bäst för varje tillfälle. Datan som skickas är i relativt små mängder och därmed väl optimerat.**

I. INTRODUKTION

Datorspel har under senare tid blivit allt mer populärt, speciellt onlinespel. Alla spelutvecklare väljer att utveckla onlinespel olika, trots att en del likheter består. För att ett onlinespel ska fungera bra så krävs det ett smidigt utbyte av data. I just onlinespel ökar kraven mer och mer på överföringshastighet. Trots att en användare har längre pingtid eller sämre överföringshastighet än andra ska det fungera smidigt ändå. I en värld där fiber blir allt vanligare så sätts även högre krav på spelutvecklarna att utveckla ett stabilt, effektivt och snabbt system som tillfredsställer användarnas behov.

Denna rapport tittar närmare på spelet Rainbow Six Siege och framförallt hur mycket paket det skickar och via vilka protokoll. Sedan följer en analys av den insamlade datan där vikt läggs vid just datamängder, säkerhet och protokollval.

II. BAKGRUND

Rainbow Six Siege är ett online skjutspel som har flera olika spellägen. Spelet är utvecklat av Ubisoft och för att köra det så krävs det att ett klientprogram är igång som heter Uplay. De aktuella spellägena i denna rapport är terrorist hunt(single player), terrorist hunt(multiplayer), situations och vanlig multiplayer. För att kunna köra Rainbow Six Siege så måste spelaren vara uppkopplad till Internet, vare sig spelaren vill spela ensam eller med/mot andra spelare. I spelläget terrorist hunt kan spelaren välja att spela ensam eller med upp till 4 andra spelare mot datorstyrda bottar. Situations är det primära single player läget men kräver ändå internetuppkoppling. Vanlig multiplayer är ett läge där 5 spelare möter 5 andra. [1]

Innan varje spelsession startas så får spelaren en stund på sig att välja utrustning och karaktär att spela som. Sedan initieras banan genom att den laddas in från hårddisk och vid denna fas väntas alla spelare in. Det finns även en designerad plats på skärmen där de andra spelarnas status visas.

Beroende på vad ett program försöker göra över Internet så kan olika protokoll vara till fördel att använda i olika situationer, detta på grund av protokollens egenskaper. I ett spel som opererar online konstant så finns det även olika tillfällen och olika spellägen i spelet där olika protokoll kan vara mer eller mindre användbara. Vid synkronisering med en spelserver är det viktigt att inga paket går förlorade. Vid pågående spel är det viktigt att alla paket kommer fram snabbt och paket förluster är acceptabelt i dessa situationer.

III. FRÅGESTÄLLNING

Syftet med arbetet är att analysera och undersöka paket som skickas respektive tas emot av spelet Rainbow Six Siege. Intressant är att analysera paket och protokoll samt att ta reda på hur det skiljer sig i de olika spellägena. Frågorna som arbetet ska besvara är följande:

- Hur skiljer sig protokoll beroende på vilka spellägen som spelas?
- Finns det anledningar till att ett visst protokoll används i vissa situationer?
- Byter klienten vilka servrar den talar med eller är det en och samma hela tiden?
- Hur mycket data skickas, vilken bandbredd används?
- Vilken säkerhet har applicerats på paketen?

IV. METODBESKRIVNING

Denna undersökning utfördes på en dator som med nätverkskabel är ansluten till en switch, till switchen är ett flertal datorer och annan utrustning ansluten. Switchen är sedan kopplad till en router som är ansluten till Internet via ViaEuropas fibernät. Routern använder NAT(Network Address Translation) då nätverksleverantören endast tillhandahåller en IP-adress(Internet Protocol). Totalt så består det lokala nätverket av 5 trådbundna enheter och 5 trådlösa enheter, under testen så begränsades detta till två trådbundna och en trådlös enhet, genom att stänga av övriga enheter eller stänga av deras anslutningar.

Det första som gjordes var att alla bakgrundsprocesser och program som var möjliga att stänga av, stängdes av på den aktuella datorn. När bakgrundsprocesser och program stängts av så startades WireShark och IP-adresser som kontaktades började noteras, för att urskilja vilka IP-adresser som tillhörde spelet och vilka som tillhörde Windows 10 eller bakgrundsprocesser som inte kunde stängas av. Nästa steg var att bygga ett filter i WireShark för att begränsa all irrelevant data. Detta gjordes genom att tillåta data endast till

och från den aktuella datorns IP-adress och exkludera alla de tidigare noterade IP-adresserna. Sedan startades Uplay upp och detta gav ett par nya IP-adresser att exkludera i filtret.

Därefter startades Rainbow Six Siege och mätningarna påbörjades. Först så togs mätdata med WireShark då spelet startades, sedan då det stängdes ned. Därefter togs data vid inaktivitet i menyn. Detta för att få en bild av hur mycket data som skickas respektive tas emot vid inaktivitet. Sedan togs mätdata vid de olika spellägena. För varje spelläge så gjordes följande: Via menyn klickades det fram till det spelläge som skulle testas, sedan startades WireShark inspelningen och därefter startades matchen. Detta gjordes för följande spellägen, single player, multiplayer och terrorist hunt (både single player och multiplayer).

För varje test så startades loggningen om i WireShark och den resulterade filen sparades, detta för att förenkla analysen av de olika momenten senare. I WireShark så gicks sedan de olika spellägena igenom för att titta närmare på pakettyp, hastigheter, hur många olika platser/servrar vi ansluter till, kommunikationens säkerhet och om vi kan se att någon onödig data skickas med.

V. VALIDITETSDISKUSSION

Den största felkällan är högst sannolikt datamängden, datan är bara insamlad under en match per speltyp. Det skulle kunna variera från match till match vilket inte har tagits hänsyn till. Hur mycket data som skickats och tagits emot är möjligen också lite inexakt eftersom många IP-adresser filtrerats bort. Då Uplays IP-adresser är bortfiltrerade kan den skickade och mottagna datamängden vara felaktig. Detta då spelet kräver tillgång till Internet och att Uplay är igång, vilket öppnar för möjligheten att data rörande spelet skickas och tas emot via Uplay klienten vilket inte har tagits hänsyn till.

En annan felkälla är om en eller flera anslutningar missats när olika IP-adresser filtrerats ut i de massiva datamängderna för respektive speltyp. Till exempel så är det 38 870 paket bara för speltypen terrorist hunt multiplayer.

Trots dessa felkällor är inte syftet med undersökning att undersöka det exakta antalet skickade respektive mottagna paket från spelet. Syftet är att besvara frågorna i sektion III, vilket fortfarande kan göras genom att undersöka den datan vi har med en rimlig noggrannhet. Eftersom det snarare är det övergripande användandet av protokoll som är relevant än om det någon enstaka gång skickas mer i något av protokollen. Då ett stort antal IP-adresser filtrerats ut går det även snabbt att märka serverbyten, även om inga sådan skedde under pågående matcher.

VI. RESULTAT

De IP-adresser som spelet var anslutet till var rätt statiska, för varje uppstart av spelet i alla fall. Det som varierade var vilken server som användes vid mulitplayerspel, den ersattes vid varje ny match. De statiska servrarna ägs av Ubisoft själva, medans de hyr servrar av Microsoft Azure till multiplayer delen. I tabell I är de fyra Ubisoft IP-adresserna listade tillsammans med de två från Microsoft Azure och de två från Vivox, som användes vid de spelen som är visualiserade i graferna.

IP	Protokoll	Land	Företag
216.98.54.23	TCP	USA	Ubisoft
216.98.62.46	TCP	USA	Ubisoft
216.98.62.72	TCP	USA	Ubisoft
216.98.55.85	UDP	USA	Ubisoft
40.68.201.140	UDP	Holland	Microsoft Azure
40.113.149.253	UDP	Holland	Microsoft Azure
74.201.107.45	RTP	USA	Vivox
74.201.107.23	RTP	USA	Vivox

TABLE I
IP-ADRESSER

I menyn när ingenting händer så skickas extremt lite paket, endast 4.77 paket/s med en bandbredd på 11Kb/s, se figur 1. UDP(User Datagram Protocol) står för 56.3% av paketen och 37% av datamängden, TCP(Transmission Control Protocol) står för 32.1% av paketen men 52% av datamängden.

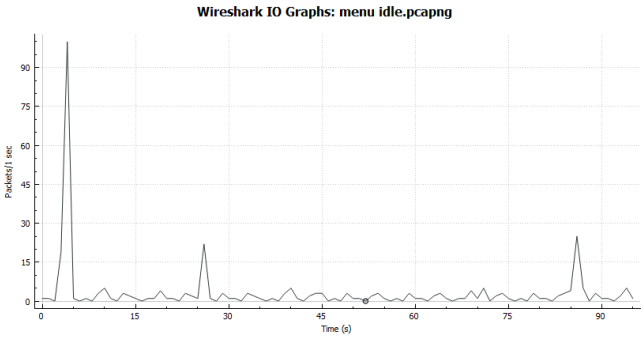

Fig. 1. Meny, overksam

I figur 2 syns det att när multiplayer laddas så skickas det färre paket då sessionen initieras. Dock så skickas det ändå en del paket, men trafiken ökar markant vid start av spelet. Det primära protokollet är UDP, men även lite TCP och RTP(Real-Time Transport Protocol, går över UDP, men är för ljud och bildöverföring i realtid, används här för röstkommunikation inom laget) observerades.

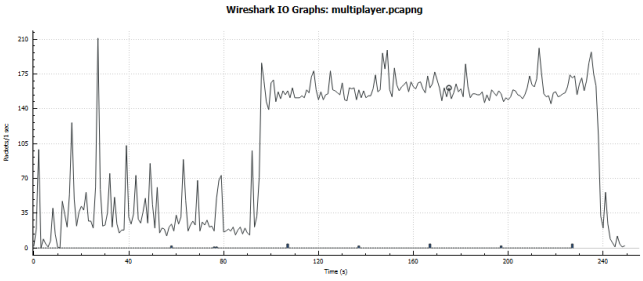

Fig. 2. Multiplayer

Paketmängden under multiplayer uppmättes till 159.03 paket/s med en bandbredd på ca 180 Kb/s. 94% av de paketen är UDP till Microsoft Azure servern, 1.8% är UDP och 1.3% är TCP till Ubisoft, RTP står för 2.2% av paketen. UDP

trafiken står för 69% av datamängden medan TCP står för 2.6% och RTP för endast 0.8%.

För single player så skickas ganska få paket men det är ändå små spikar då och då, se figur 3. De stora spikarna i början och slutet är när banan startas och avslutas. Även här är det mestadels UDP men inte lika stor andel, 52.2% av alla paket, men endast 28.4% av datamängden, TCP står för 37.2% av paketen men 55.8 % av datamängden. Det är också avsevärt mindre data som skickas i single player, 5.96 paket/s och 14 Kb/s. Anslutningarna är vidare endast till Ubisofts egna servrar.

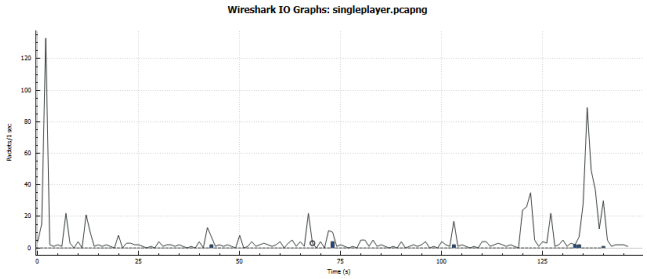

Fig. 3. Singleplayer, även kallat situations

I terrorist hunt multiplayer så är det, som i vanliga multiplayer, nästan uteslutande UDP till Microsoft Azure, 92%. Resten består av en liten del UDP paket, 2.8%, och TCP paket, 2%, till Ubisofts servrar, och en liten del RTP paket, 2.5%. RTP står för 0.6% av datamängden, medan UDP står för 60.7% och TCP 6.1%. Den totala data mängden är bara hälften av vanliga multiplayer, 72.9 paket/s och en bandbredd på ca 98Kb/s. I figur 4 syns tydligt när matchen väl startar runt 80 sekunder.

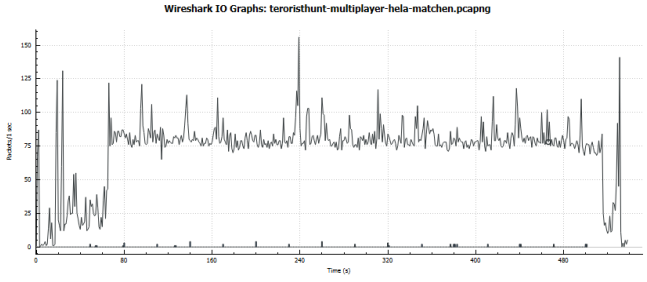

Fig. 4. Terrorist hunt, mulitplayer

Det sista spelläget som testades var Terrorist hunt som ensam spelare, här är det som i vanliga single player inte mycket data som skickas, 5.92 paket/s med en bandbredd på 10Kb/s. UDP står för 37.2% av paketen och 31.6% av datamängden medan TCP står för 23% av paketen och 44.3% av datamängden, detta till Ubisofts servrar. Lite förvånande så skickades även en hel del RTP paket, 30.3% av paketen och 4.5% av datamängden, till Vivox, det bolag Ubisoft använder för röstkomunikation i spelet.

VII. DISKUSSION

All kommunikation i spelet sker med centrala servrar, detta har troligen flera olika anledningar, så som att Ubisoft vill

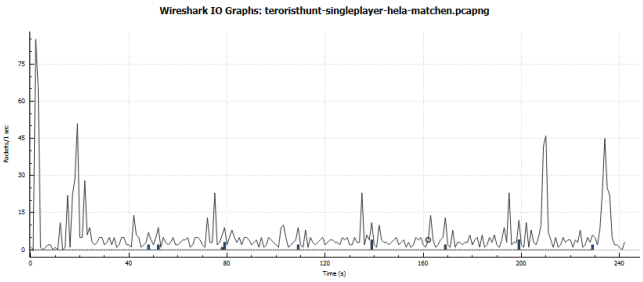

Fig. 5. Terrorist hunt, singleplayer

behålla så stor kontroll som möjligt över spelet, men det skyddar även de spelande från till exempel DoS(Denial of Service) attacker, genom att ingen får ta del av de andra spelarnas IP-adresser. [2]

Att Ubisoft hyr in servrar för kommunikationen under multiplayer spel är troligen för att snabbare kunna skala upp och ner utifrån hur många som spelar, till exempel inför en uppdatering med nya banor kommer ut.

Eftersom att Rainbow Six Siege är ett onlinespel så är det primära protokollet UDP det enda rimliga valet för merparten av all data som skickas under aktivt spelande med andra. Men även TCP och RTP används i multiplayer lägena, RTP till röstkommunikationen mellan lagmedlemmarna medan det exakta syftet med TCP ej har kunnat säkerställas då all trafik är krypterad med TLS v1.2. Det misstänks att det inte är relaterat till det aktiva spelandet utan olika former av bakgrundsinformation i spelet, till exempel uppdateringar av vad man åstadkommit eller vem som vann eller hur många man sköt i en match.

Det är lite synd att spelutvecklarna inte tagit steget över till TLS v1.3 som publicerades i augusti i år, mer än fyra år efter det första utkastet. Men detta antas vara för att spelet är ett par år gammalt och att TLS v1.2 fortfarande anses vara rätt säkert. [3]

Vilken säkerhet som har applicerats på UDP, och därmed RTP, har inte kunnat fastställas mer än att all data verkar krypterad, ingen information har heller kunnat säkerställas angående detta vid internetsökningar. Detta beror troligen på att Ubisoft inte sprider den informationen då en viss nivå av säkerhet ges av att en eventuell angripare nu först måste tyda hur de krypterat datan innan ett försök att avkryptera den kan inledas.

Likaså är det svårt att sia om någon otillbörlig data skickas iväg av spelet, men inga konstiga anslutningar fanns, så om de skickar data de inte bör så gör de det åtminstone till sina egna servrar.

Att datamängden i terrorist hunt multiplayer är hälften av den i vanliga multiplayer är rimligt då spelläget endast har ett lag med upp till 5 riktiga spelare, istället för två lag, vilket kraftigt minskar mängden data som behöver delas.

Att RTP skickas i de båda multiplayerlägena känns rimligt då röstkommunikation mottogs från andra spelare i dessa. Men att det görs i terrorist hunt single player läget verkar väldigt udda, det fanns heller ingen mikrofon ansluten som skulle kunnat ta upp ljud att skicka till servern. Den enda

förklaringen som kan ses är att de två terrorist hunt lägena egentligen är ett och samma men utan medspelare i single playerläget. Där av så skickas liknande mängder RTP även i single player, men att detta framstår som en mycket större mängd då ingen multiplayer UDP trafik förekommer.

De saknade procenten för att datan skall bli 100% är overhead så som ethernet,- och IP-headers och några enstaka ARP-requests.

VIII. SLUTSATSER

UDP är som förväntat det dominerande protokollet för aktivt spelande med andra. Mer förvånande är att datamängden är så begränsad som den är, det visar verkligen på hur optimerad den datan som behöver delas för en bra spelupplevelse blivit. Det öppnar i sig upp för en större marknad med kunder, alla de som inte har så bra Internet uppkopplingar som vi i Sverige.

Säkerheten bedöms som god, UDP är säkerställt via en kryptering som inte kunnat identifieras och är således omöjlig att betygsätta, dock så finns inga rapporter om några säkerhetsintrång. Utrymme för förbättringar föreligger dock för TCP med en uppgradering till TLS v1.3 för alla anslutningar.

IX. FORTSATT ARBETE

Delar att titta vidare på är skillnader mellan det som gåtts igenom i den här rapporten och när en egen server sätts upp. Detta är inte som en del spel, framförallt äldre, där den egna servern kördes på egen hårdvara, utan det sköts fortfarande av Ubisoft. Den kontroll som ges är istället vilka man spelar med, vilken bana och vilka regler som gäller.

En annan sak att titta mer på är hur Uplays data ser ut och om den varierar mellan skrivbordet, startande av spel och under spelande i de olika spellägena. Detta för att göra en bedömning om även Uplay applikationen skickar spelrelaterad data under spelande.

REFERENSER

[1] Wikipedia, Rainbow Six Siege, 13/12-18, https://sv.wikipedia.org/wiki/Tom_Clancy's_Rainbow_Six_Siege

[2] Wikipedia, DoS, 13/12-18, https://sv.wikipedia.org/wiki/Denial_of_Service

[3] Wikipedia, TLS, 13/12-18, https://sv.wikipedia.org/wiki/Transport_Layer_Security