Undersökning av Discord med Wireshark

Anders Klint
Lunds Tekninska Högskola
Email: dat15akl@student.lu.se

Vilhelm Åkerström
Lunds Tekninska Högskola
Email: dat15vak@student.lu.se

***Sammanfattning*—VoIP-program blir allt mer populära. Men många av dessa applikationer börjar bli gamla och har ett flertal problem. Ett modernt alternativ är Discord, en VoIP-applikation under utveckling. Denna applikation har undersökts med hjälp av paktetsniffaren Wireshark. Undersökningen kollade på vilka datapaket som skickades vid inlogging, textmeddelanden och röstsamtal. Förutom detta undersöktes även om paketen först skickades till en lokal server eller om de gick direkt till mottagaren och hur stor bandbredd som användes i passivt respektive aktivt läge. Det visade sig att Discord är en Client-Server-applikation och den skickar UDP och TCP-paket beroende på vilken typ av kommunikation som används. Detta då de olika protokollen är olika när det gäller hur snabbt och säkert de kan skicka information. Discord använde UDP för röstsamtal och använde även protokollet TLS för säker kommunikation vid textmeddelanden och inloggning. TCP var till för bekräftelser att TLS-paketen tagits emot.**

I. Introduktion

Användningen av VoIP eller Voice over IP växer allt mer. VoIP är ett sätt att kommunicera i samtal eller textchatt via Internet. Applikationer som Skype och TeamSpeak är exempel och blir allt mer populära att använda. Till exempel har Skype över 300 miljoner aktiva användare varje månad [1]. Dessa applikationer börjar dock bli gamla, Skype är mer än 10 år gammalt och har under den tiden bland annat bytt från ett peer-to-peer baserat system till ett client-server baserat system [4]. För ungefär ett år sedan fick de dock en ny konkurrent i form av Discord [2]. Discord är ett VoIP-program som släpptes 2015 av före detta spelföretaget Hammer & Chisel. Deras idé var att bygga en VoIP-applikation för en ny generation.

Discord är från grunden byggd att vara gjord för kommunikation mellan folk som spelar datorspel[3]. Detta betyder att den är byggd för att vara så latensfri som möjligt och även kräva så lite processorkapacitet som möjligt, så att voice chat ska gå snabbt och så att processorn kan användas till spelet. Det är dock en applikation under utveckling, vilket betyder att det fortfarande saknas viss funktionalitet som är vanligt i många andra VoIP-program. För tillfället finns det möjlighet att skicka meddelanden och samtala direkt, men även sätta upp sina egna servrar och skicka meddelanden och samtala via dessa, men det finns inga möjligheter att ha videosamtal.

En annan viktig funktion för VoIP är säkerheten. Skype har haft stora problem med sin användarsäkerhet[4]. Därför har Discord satsat på säkerheten och har även ett program där vem som helst får försöka hitta buggar och säkerhetsrisker i programmet[5].

Med hjälp av paketsniffaren Wireshark ska hur och varifrån Discord skickar datan mellan användare undersökas. Undersökning kommer innefatta vilka paket som skicka vid inloggning, direktmeddelanden och röstkommunikation.

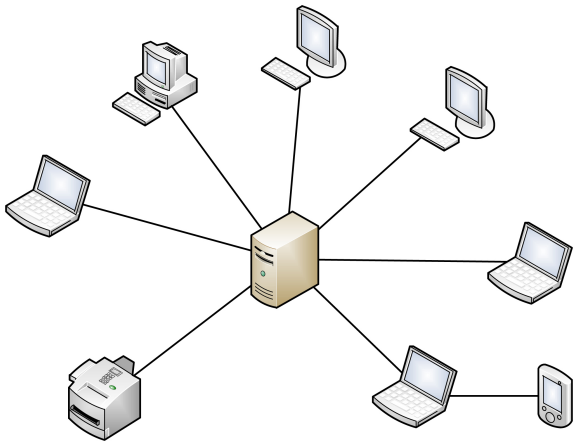
II. Teori

A. Setup

För detta experiment användes en dator med IPv4: 192.168.0.30 och en External IP: 83.254.129.252 kopplad med en Ethernet-kabel till en router med IPv4: 192.160.0.1. Nätverkskortet i datorn var av typen Realtek 8821ae. Alla aktiva applikationer som fanns på datorn stängdes av utom Windows-processer.

B. Client-Server:

Client-Server-modellen innebär att all utgående data skickas till en serverdator, som sedan tolkar och behandlar datan på ett specifikt sätt och skickar vidare den till rätt destination. Detta betyder att två datorer som kommunicerar inte vet mer om varandra än den data som skickas. T.ex. betyder detta att en dator inte kan få tag på en annan dators IP om inte servern tillåter det. I *figur 1* visas en illustration som exempel på hur ett Client-Server-nätverk kan vara uppbyggt.



Figur 1 - Ett Client-Server-nätverk [8]

Discord använder sig av Client-Server. Det finns servrar utspridda i hela världen, och användaren har friheten att välja server själv.

C. TLS & TCP:

Discord använder sig av TCP som står för Transport Layer Protocol, vilket är ett vanligt dataöverföringsprotokoll och

används i huvuddelen av all internetkommunikation[7]. TCP tillhandahåller en pålitlig ansluting mellan datorer och har inbyggda felkorrigerande egenskaper som är viktiga när data måste komma fram intakt.

Discord använder sig av TLSv1.2 som står för Transport Layer Security och används för att skicka textmeddelanden. TCP används för konfirmation av att dessa TLS-paket har kommit fram. TLS är ett lager över TCP som används för säker kommunikation över nätet[6]. Detta genom att kryptera all data som skickas genom en nyckel som förhandlas fram vid upprättandet av en länk. TLS används både när en användare skickar meddelanden direkt via Discord och när en egen server används.

D. UDP:

När det kommer till röstsamtal använder sig Discord av UDP (User Datagram Protocol). UDP är ett snabbare protokoll då det är förbindelselöst, men det innebär också att det inte finns något sätt att kolla om paketen kommer fram, eller att mottagaren har fått dem i rätt ordning [7].

E. Specifikt för Discord

Discord i webbläsaren använder WebRTC som är till för kommunikation, men gör det snabbare genom att få webbläsaren att tro att det är peer-to-peer som används när den ansluter till Discords servrar[2].Detta betyder att Discord använder mindre processorkraft, eftersom att den tekniskt sett bara kopplar sig till en enda peer.

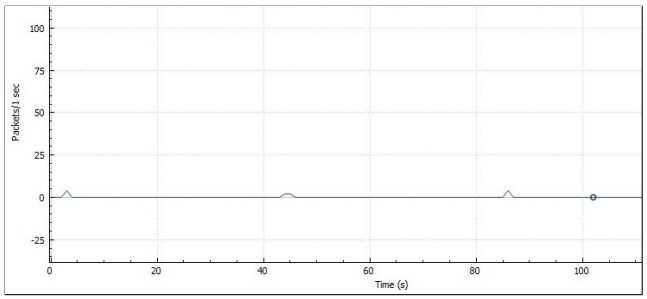
III. Resultat

Efter att ha undersökt IP-paketen som skickas och tas emot av Discord visar det sig att all kommunikation sker mellan servrar, vilket leder till att inget spår av andra klienters IP-adresser går att hitta. I undersökningen valdes Centraleuropa som server. Efter att ha sniffat med Wireshark under ett VoIP-samtal mellan två klienter hittades serverns IP-adress. Sedan användes hemsidan *IPLocation.net* för att hitta information om servern från dess IP-adress. Servern visade sig ligga i Frankrike, Roubaix, och var hostad av OVH-SAS - ett hostföretag för dedikerade servrar och cloudtjänster. Vidare märktes att UDP-paket användes för röstkommunikationen, men TCP-paket användes för att *pinga* klienterna innan röstsamtal, d.v.s. fråga servern om motparten är online. För ett mer exakt resultat kunde nu datatrafiken filtreras i Wireshark med de IP-adresser Discord använde. I *tabell 1* visas samtliga adresser med deras relaterade information.

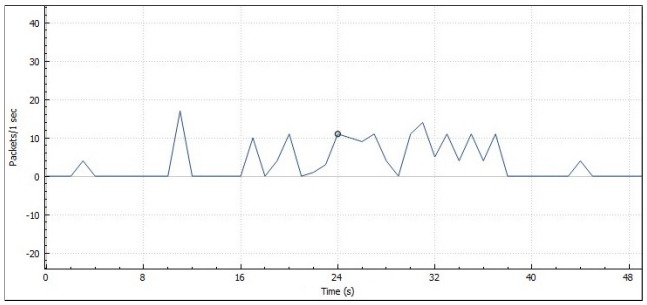
Därefter undersöktes hur mycket bandbredd Discord använde vid passivt samt aktivt tillstånd. I passivt tillstånd skickades minimal data, endast ett litet antal (ca 4 st) TCP-paket var 45:e sekund *(se figur 2.1).*

I det aktiva tillståndet undersöktes bandbredden för att skicka meddelanden, samt ett öppet röstsamtal. I *figur 2.2* visas paketen för 10 skickade meddelanden under en kort tid. Här användes endast TCP-paket.

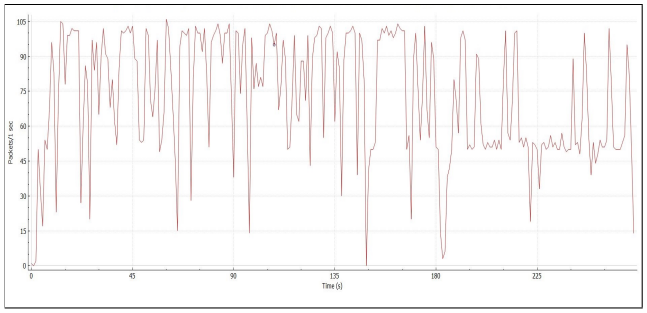
Röstsamtal använde däremot endast UDP-paket för dataöverföring, visualiserat i *figur 2.3.*. En titt på *figur 2.3, paketöverföringen för röstsamtal,* visar en maxhastighet på ca 100 paket per sekund. Där varje paket är ungefär mellan 150-250 bytes stort.



figur 2.1 - Figuren visar Discords bandbreddsanvändning i passivt läge när inga meddelanden skickas.



figur 2.2 - Bandbreddsanvändningen av Discord när 10 textmeddelanden skickades, TCP-paket.



figur 2.3 - Bandbreddsanvändningen av Discord under röstsamtal, UDP-paket.

Vidare undersöktes hur direktmeddelanden skickades, både på en Discord-server och direkt till en annan användare. Dessa tycktes vara oberoende av vilken server användaren valt, då all information skickades till ett antal servrar i Californien, varav alla var hostade av *CloudFlare*. Protokollet för paketen var TCP, där datan som skickades; självaste meddelandet användaren skrev, var krypterad.

Till sist testades vilka paket som skickades vid inloggning. Som textmeddelanden kommunicerade inloggningen också endast med CloudFlare-servrarna i Californien. Här skickades dock TCP-paket krypterade med TLSv1 och TLSv1.2.

IP-adress	Företagd	Plats	Syfte
104.16.58.5 104.16.59.5 104.16.204.240	CloudFlare	Californien	Cloud - inloggning & meddelanden
149.202.70.136	Discord	Frankrike	Röstkommunikation

tabell 1 - Mappning av de IP-adresser Discord använde sig av.

IV. Slutsats

A. Validitetshot

Under undersökningen var det flera faktorer som kan ha påverkat resultatet. Ett av dessa är bakgrundsdata som skickades från datorn. För att bli av med den filtrerades vilka paket som visades av Wireshark. Denna filtrering gjordes så att bara en typ av protokoll visades. Detta innebar dock att om Discord skickar andra typer av paket så syns inte detta och om någon annan bakgrundsprocess skickar samma sorts paket så kommer även dessa visas. Därför observerades först vilka typer av paket som skickades av Discord i olika lägen och sedan lades filter till efter detta. Tyvärr kan det fortfarande komma paket av samma typ från bakgrundsprocesser, men de sågs som försumbara.

B. Servrar

Client-Server-modellen är väldigt resurskrävande för företaget som hostar servrarna, då all data måste processeras via dessa servrar. Däremot är det väldigt fördelaktigt för användaren, då det inte finns några direktkopplingar mellan användarna och därmed gör dem osynliga för varandra. Detta leder till att användarnas IP-adresser hålls gömda, och ökar därmed säkerheten. En annan fördel är att två användare inte måste vara online samtidigt för att kunna skicka meddelanden till varandra. Om peer-to-peer-modellen använts skulle meddelanden endast kunna skickas när båda användare är online samtidigt. Till sist minskar Client-Server-modellen stressen på användarens nätverk vid röstsamtal med fler än en person. Istället för att användaren behöver skicka data till varje peer, skickas endast data till servern, sedan skickar servern datan till samtliga peers. Detta är dock en anledning varför denna modell gör det väldigt resurskrävande för företaget.

I Discord fanns det alternativ för användaren att välja server själv. Förutsatt att användaren väljer den server närmast både sig själv och sina peers den kommunicerar med, blir latensen så låg som möjligt, då den fysiska vägen för dataöverföring blir mindre. Därför har Discord hyrt servrar över hela världen. Det märktes dock att dessa servrar endast användes vid röstkommunikation. Detta troligtvis eftersom, som ovannämnt, latensen endast är viktigt vid röstkommunikation. Dessutom är det vid röstkommunikation som den största mängden data transporteras, vilket jämförelsen mellan bandbreddsgraferna i resultat-delan av rapporten visade. All annan kommunikation som chattfunktionen och inloggning kan därför ske med en huvudserver placerad var som helst, i detta fall Californien. Denna server var hostad av företaget CloudFlare. Eftersom all chatthistorik sparas, och att de meddelanden som missats när användaren varit offline inte raderas, kan slutsatsen dras att Discord använder CloudFlares servrar som cloudplattformer.

C. Paket

Användning av UDP-paket vid röstsamtal ger låg latens, vilket är oerhört viktigt vid röstkommunikation. Speciellt bland gamers, vilka Discord för det mesta marknadsför sig till. Vid paketförluster slängs dock paketen bort, och informationen forsvinner helt. Om istället TCP-paket använts skulle motparten vara tvungen att vänta tills paketet kommit fram vid paketförlust. Detta är inte önskat vid röstkommunikation, då det vid varje paketförlust skulle skapa längre och längre fördröjning mellan parterna. Bortkastningen av paket spelar inte heller särskilt stor roll i direktsamtal - motparten kan enkelt upprepa den information som gick till miste. Därför verkar UDP vara optimalt vid röstsamtal.

Vid chattkommunikationen användes däremot TCP-paket, vilket är passande då latens inte är ett stort problem vid textkommunikation. Här har däremot paketförluster stora konsekvenser, då korrupt textdata är mycket svårare att avtyda än korrupt ljuddata - ljuddatan är större så förluster på ett visst antal bytes spelar mindre roll jämfört med textdata. Därför används TCP-paket för en stabil dataöverföring.

Vid inlogging behövs också en stabil dataöverföring för att förhindra fel vid verifieringsprocessen *(eng. authentication process)*. Paketförluster här gör att inloggningen misslyckas, och användaren måste försöka igen.

D. Bandbredd

Discord marknadsför sig som resurssparande, vilket sett från ett bandbreddsperspektiv verkar stämma då Discord vid passivt läge skickade minimal data. Vid dataöverföring som chattmeddelanden och röstkommunikation verka ingen onödig data skickas, utan endast den väsentliga. Dataöverföringshastigheten var också godtyckligt snabb. Detta leder till relativt hög ljudkvalite vid röstsamtal.

E. Säkerhet

Då Client-Server-modellen och TLS-paketen vid inloggning ökar säkerhet, märktes även att det inte gick att få ut någon vettig information ur paketen som innehöll chattmeddelanden. Förmodligen gör Discord en lokal kryptering av datan innan den skickas, vilket leder till att ingen som har åtkomst till nätverket användaren sitter på kan läsa informationen. Om samma sak gäller för röstkommunikationen är svårt att säga, då det i denna undersökning inte gjordes någon avkodning av den transporterade ljuddatan. Säker kryptering av data kräver mycket processorkraft [9]. För textdata är det inget problem då den är så liten, men för ljuddata kan det blir problem. Om ljuddatan ska krypteras kommer det inte bara öka latensen mellan användarna, utan även kräva stor processorkraft från både servrarna och klienterna. Eftersom Discord marknadsför sig som resursvänligt [10] är det inte troligt att röstsamtal krypteras. Det skulle alltså ta upp för mycket processorkraft för användaren, och ännu mer för servrarna. Om detta är fallet finns det en säkerhetsbrist, då en paketsniffare enkelt

kan lyssna på röstsamtalen. Däremot kan ingen slutsats om detta dras utan vidare undersökning.

REFERENSER

[1] Surur, *https://mspoweruser.com/skype-300-million-monthly-active-users*,Skype has more than 300 million monthly active users, will get bots Mar 2016, Dec 2016

[2] T. Marks, (http://www.pcgamer.com/one-year-after-its-launch-discord-is-the-best-voip-service-available), One year after its launch, Discord is the best VoIP service available. (Maj 2016), Dec 2016

[3] *https://discordapp.com/company*, Our Point of View, Dec 2016

[4] T. Warren, *http://www.theverge.com/2016/11/8/13561024/microsoft-skype-baidu-linkedin-hack*, Why are Skype accounts getting hacked so easily?. Nov 2016. Dec 2016

[5] Team Discord, *https://discordapp.com/security*, Bug bounty discord & co.. Dec 2016

[6] T. Dierks; E. Rescorla (August 2008). The Transport Layer Security (TLS) Protocol, Version 1.2, Dec 2016

[7] M. Kihl, J. Andersson, *Datakommunikation och nätverk*, Studentlitteratur, 2015

[8] *https://itpeernetwork.intel.com/top-10-reasons-to-setup-a-client-server-network/*, Bildkälla: Top 10 Reasons to Setup a Client-Server Network, Dec 2016

[9] *https://www.google.com/patents/US5751813*, Use of an encryption server for encrypting messages , Dec 2016

[10] Team Discord, *https://discordapp.com/features*, See How Discord Stacks Up. Dec 2016