# Säkerhet i peer-to-peer-system

# Sammandrag

Peer-to-peer-nätverk har på senare tid fått stort genomslag och som när det gäller andra mjukvaruapplikationer innebär detta säkerhetsproblem. Integritet är en viktig aspekt när det gäller säkerhet, både integriteten i datan och applikationerna, och inom detta område finns flera problem för p2p-nätverk som i slutändan bottnar i tillit inom nätverket. Hashning och kontroll och testning av applikationerna är metoder för att förbättra integriteten.

En annan viktig del ur säkerhetssynpunkt är sekretess, något som gäller både data och användare i ett p2p-system. Kryptering är ett viktigt verktyg för att säkerhetsställa sekretessen för kommunikationen i nätverket och datan, och för att skydda den enskilde användarens personliga integritet finns idéer såsom FreeNets.

Det finns även flera olika typer av attacker mot p2p-nätverk. Denial of service-attacker kan genomföras på flera olika sätt beroende på vilken typ av p2p-nätverk det är och hur det är uppbyggt.

# Inledning

Kommunikationsparadigmet peer-to-peer, hädanefter refererad till som p2p, har på senare tid fått stort genomslag i olika datanätverk, inte minst Internet. Den största bidragande faktorn till teknikens numera stora utbredning kan främst tillskrivas de fildelningsapplikationer som introducerades och fick sitt genombrott på marknaden i början vid millennieskiftet, med exempelvis Napster som en av de första. Även inom andra områden än fildelning har p2p gått starkt framåt, t.ex. med större distribuerade beräkningsnätverk som använder tekniken såsom SETI@home.

Som med alla nya tekniker, kanske främst när det kommer till nätverk och applikationer och protokoll inom området, så är säkerheten inte det som kommer i första hand när man konstruerar och lanserar de första versionerna. På grund av p2p-nätverkens snabba och stora utbredning har detta inneburit att en stor mängd användare har fått tillgång till och börjat använda applikationer och tekniker som inte har blivit testade fullt ut ur säkerhetssynpunkt.

## *Syfte*

Syftet med den här rapporten är att titta på säkerhet och problem i p2p-system. Vi har valt att fokusera på säkerhet i p2p-nätverk ur tre synpunkter: Integritet och tillit, sekretess och användares personliga integritet samt attacker mot peer-to-peer-nät. Inom dessa områden undersöker och exemplifierar vi själva problemen. Vi tittar på orsaker till problemen, och huruvida det finns lösningar till dem och i så fall hur de fungerar. Både interna och externa problem med säkerheten i systemen tas upp.

Rapporten är strukturerad som följer: först behandlar vi integritet och tillit i p2p-system. Därpå följer sekretess och användares personliga integritet följt av attacker mot peer-to-peer-nät. Avslutningsvis har vi en sammanfattning och en kortfattad slutsats.

# Integritet och tillit

På grund av p2p-nätverks decentraliserade struktur och öppna filosofi är integriteten inom nätverket en kritisk punkt när det gäller säkerheten i dessa nät. Detta gäller främst integriteten i den data nätverket hanterar, exempelvis filer i fildelningsnätverk eller beräkningsparametrar och –resultat i distribuerade beräkningar. Även de applikationer som används för att ansluta till och använda nätverket utgör en riskfaktor när det gäller integriteten.

Problemet med p2p-nät i dess renaste form är att eftersom det inte finns någon central server i systemet så krävs att operationerna i nätverket administreras i distribuerad form och gemensamt utav dess deltagare. Detta gör att det heller inte finns någon som är centralt ansvarig för integriteten i nätverket. Inte heller i hybridformer av p2p-nät är

integriteten något som är lätt att kontrollera. Detta beror främst på att alla deltagare bidrar till nätverket och att hålla ordning på integriteten i de enskilda noderna är problematiskt.

I följande del exemplifierar vi de olika problematiska delarna och tittar på olika lösningar till problemet, samt tittar på en gemensam nämnare för de olika problemen: tillit.

## *Integritet i applikationer*

För att kunna ansluta till och utnyttja p2p-nätverk krävs idag speciella applikationer vilka kan utgöra en kritisk aspekt när det gäller nätverkets integritet. För att kunna lita på nätverkets funktionalitet krävs att man litar på den applikation som utnyttjar nätverket, och skulle applikationen nyttja nätet med ont uppsåt äventyrar det hela nätverkets integritet och funktionalitet.

Ett exempel på hur en applikation kan bete sig felaktigt och påverka funktionaliteten är att i ett fildelningsnätverk bara utnyttja möjligheten att komma åt andra noders filer utan att bidra med egna. Detta är något som strider mot filosofin för ett fildelningsnätverk, och har en teoretisk möjlighet att stoppa nätet från att fungera helt. Skulle alla klienter vara sådana som inte delar med sig så skulle det inte finnas någon data att ta del av.

En lösning på detta är att bygga in funktionalitet i själva nätverket som hindrar eller begränsar möjligheter för de klienter som beter sig illa. Exempelvis har BitTorrents klientmjukvara den funktionaliteten att ju bättre ens uppladdningshastighet är, desto bättre är nedladdningshastigheten[1]. Det gör att eventuella användare som endast tankar får väldigt dålig eller ingen funktionalitet alls från nätverket.

Ett annat problem när det gäller applikationers integritet är när applikationen, utan användarens kännedom, innehåller funktionalitet som skadar nätverket eller den enskilda noden. Det kanske mest kända exemplet på detta är fildelningsapplikationen Kazaa, som visade sig installera spyware tillsammans med p2p-applikationen[2] på användarens dator, ofta utan att användaren visste om detta.

En lösning på detta problem är att undersöka och verifiera applikationens funktionalitet, och är applikationens källkod öppen kan man analysera denna för att säkerställa huruvida applikationen beter sig på rätt sätt. Skulle mjukvaran vara proprietär kan inte källkoden verifieras av utomstående och verifieringen av mjukvaran försvåras. I slutändan kan detta leda till att det enda som kan avgöra om man använder applikationen eller inte är huruvida man har tillit till mjukvarans skapare.

## *Integritet i data*

Mer kritiskt än applikationerna är den data p2p-nätverket använder sig av. Detta beror främst på att datan i ett p2p-nätverk är det centrala i nätverket; ett fildelningsnätverk är ingenting utan filerna, och i distribuerade beräkningar är det parametrarna och resultaten som är de viktiga delarna. I dessa båda typer av nätverk är det viktigt att man kan lita på att den data man som deltagare får tillgång till är korrekt.

Det kanske mest påtagliga när det gäller datans integritet i fildelningsnätverk är huruvida filerna innehåller det de utger sig för att innehålla. Om man har bara ett filnamn att gå på kan filen i fråga innehålla vad som helst. Exempelvis har virus och annan malware spridit sig i fildelningsnätverk[4] genom att filer innehållande malware har laddats ner av användare i tron att det de laddar ner är något annat. Vidare har fildelningsnätverk drabbats av så kallade *decoy files*, falska filer, vars enda syfte är att försvåra för användaren att hitta det de söker efter.

Även beräkningsnätverk drabbas av integritetsproblem. En av de största distribuerade beräkningsapplikationerna, SETI@home, drabbas av att användare fuskar sig till bättre status i beräkningsrankningen genom att generera falska data[4]. Även om SETI@home inte är ett rent p2p-nätverk i dess rätta bemärkelse så är det uppenbart att liknande problem kan drabba p2p-implementationer av beräkningsnätverk.

Lösningen på integritetsproblemen när det gäller datan i p2p-nätverk är även här att verifiera datan och dess integritet. Det vanligaste sättet att göra detta i fildelningsnätverk är att använda sig av olika hashfunktioner som beräknar en hash utav den fil som delas ut, och sedan delas även denna hash för att användare på egen hand ska kunna verifiera att datan är korrekt. En variant är även att hashfunktionaliteten finns inbyggd i mjukvaran som utgör p2p-nätverket. Exempelvis har BitTorrent inbyggd hashning som automatiskt verifierar datans integritet[5].

Även beräkningsapplikationer kan använda sig av hashning och checksummor för att verifiera de resultat som noderna producerar. För att göra detta krävs dock att man kan verifiera att även checksummorna är korrekta. En lösning på det är att distribuera samma data till flera beräkningsnoder för att på så sätt uppnå redundans i systemet och därigenom ha möjlighet att verifiera att samma resultat fås från oberoende källor. Detta kan även tillämpas i andra typer av p2p-nätverk, att man genom redundans kan verifiera integriteten i datan.

Ett problem som dock inte kan avhjälpas med redundans gäller filer i fildelningsnätverk där noderna själva bidrar med filer. I en sådan situation har alla filer en enskild ursprunglig källa där den första verifieringen av filen genomförs. För att sedan uppnå redundans sprids filen från den ursprungliga källan. Dock är alla fortsatta verifieringar av filen beroende på den första, vilket innebär att man måste ha tillit till att den användare och nod som publicerade filen på nätet.

## *Tillit i p2p-nätverk*

Som synes har både integritet i data och i applikationer en gemensam nämnare i slutändan: tillit. Detta problem finns självklart även i andra nätverk, men är förmodligen inte lika stor faktor som det är i ett p2p-nätverk. Har man en central server som distribuerar filer utan möjlighet för utomstående att bidra till innehållet så krävs det endast att man har tillit till den person, organisation eller det företag som står bakom servern och tillhandahåller servicen. För att generalisera kan man säga att så fort man har en central auktoritet som kan verifiera och garantera datans integritet blir det enklare att hantera tilliten eftersom man då endast behöver lita på auktoriteten.

Ett sätt att begränsa detta problem är att införa autentisering av ett p2p-nätverks användare för att kunna hålla reda på vilka användare och noder som utför vilka operationer i nätverket och på det sättet öka tilliten utav datan och användarna eftersom man vet att man kan spåra eventuella användare som agerar med ont uppsåt. Detta medför dock att sekretessen och användares personliga integritet försvinner, något som också medför säkerhetsproblem.

## Sekretess och användares personliga integritet

Ett annat problem när det gäller säkerhet i p2p-nätverk gäller sekretess och användares personliga integritet och anonymitet. Även om de flesta p2p-nätverk idag är relativt öppna kan det finnas en önskan att skydda innehållet och deltagarna från både utomstående och andra deltagare i samma nät. Det kan röra sig om att beräkningarna i ett beräkningssystem måste hållas hemliga eller att användarna i ett fildelningsnätverk ska kunna vara anonyma och att det inte ska kunna gå att spåra vilka filer de har laddat ner eller delat med sig.

Oförsiktiga användare som använder fildelningsprogram på t.ex. ett företag kan även av misstag råka dela ut känslig data utan att veta om det. Många fildelningsprogram kräver nämligen att man delar ut en viss mängd information, en användare kan då för att få ihop tillräcklig mängd data välja att dela ut hela hårddisken. Det kan leda till att lösenord eller företagshemligheter blir tillgängliga för vem som helst, trots att företaget egentligen är skyddat av brandväggar som ska hindra intrång.

I följande avsnitt går vi igenom kryptering i allmänhet som ett sätt att lösa problemet med sekretess i ett p2p-nätverk, samt redogör för FreeNet, en p2p-implementation som garanterar användares anonymitet.

### *Kryptering*

Det självklara sättet att bibehålla sekretess i ett system som nyttjar ett mer eller mindre publikt nät är att använda sig av kryptering. I fallet med p2p-nätverk finns i princip två olika aspekter där kryptering kan tillämpas för att främja sekretess. Den första och kanske minst viktiga av de två är kryptering av själva datan i nätverket.

I ett publikt fildelningsnät spelar förmodligen kryptering av datan mindre roll, just eftersom det är publikt, och själva poängen med nätet är att dela med sig utav den data man har. Dock kan man tänka sig en situation där man vill skapa grupper inom nätverket där endast vissa specifika noder får delta. I ett sådant fall kan kryptering användas, och deltagarna i gruppen får då ta del av de nycklar som krävs för att läsa datan.

Även i beräkningsnätverk kan kryptering ha betydelse. Vill man kunna utnyttja den extra beräkningskraften i ett distribuerat nät utan att publicera de parametrar och resultat beräkningarna genererar kan kryptering vara en lösning. Dock blir det problem på klientsidan eftersom beräkningarna kräver att datan dekrypteras för att kunna användas, och även om detta endast sker i minnet på noden så är det en brist i säkerheten[4].

Det andra och mer betydande av krypteringstillämpningarna för p2p-nätverk är kryptering av kommunikationen inom nätverket, till exempel med hjälp av SSL/TLS[6]. Detta har den fördelen att kommunikationen är skyddad både från utomstående och även från andra noder inom nätverket. Endast de noder som är en del av den aktuella anslutningen kan dekryptera kommunikationen. Kryptering av kommunikationen kan med fördel användas både i fildelnings- och beräkningsnätverk.

### *FreeNet*

En viktig aspekt när det kommer till användare av ett p2p-nätverk gäller bibehållandet av användarens personliga integritet. Problemet när det gäller Internet och publika nät i allmänhet är att anonymitet är svårt att åstadkomma och därmed kan alla aktiviteter mer eller mindre spåras till den användare som utfört aktiviteten. Samma problem med den personliga integriteten gäller p2p-nätverk i allmänhet och fildelningsnätverk i synnerhet.

En lösning på detta problem är FreeNet[7], ett p2p-nätverk där användarna är helt anonyma. Målet med FreeNet är att låta vem som helst publicera och läsa information helt anonymt. Detta uppnås genom att noderna i FreeNet-nätverket inte vet var den data som publiceras i slutändan lagras i nätverket, samt att enskilda noder inte vet vilken data de själva delar ut på nätverket. Vid publicering av information på nätverket tilldelas informationen ett unikt id, och datan distribueras sedan ut över nätet för att bilda kluster inom nätverket där data med liknande id lagras i närheten av varandra, detta för att underlätta sökning. Datan blir även krypterad just för att den enskilda noden inte ska veta vad den själv delar ut, och en fil kan också delas i flera delar och lagras på flera olika noder för att ytterligare öka anonymiteten.

Det finns dock ett problem med FreeNet, och anonymitet i p2p-nätverk i allmänhet, nämligen att eftersom alla noder i nätverket blir anonyma så blir även noder med ont uppsåt anonyma[8]. Det går alltså inte att spåra noder som sprider felaktiga eller malware-smittade filer i ett helt anonymt nätverk, men det är snarare ett problem med anonymiteten som koncept snarare än ett tekniskt problem i nätverket.

## Attacker mot peer-to-peer-nät

Ett p2p-nätverk är sårbart för flera olika typer av attacker mot nätverket. Följande sektioner kommer att gå igenom ”denial-of-service”-attacker samt hur man kan lösa problem med falska noder genom att använda den bysantinska överenskommelsen.

### *Denial of service*

En ”denial-of-service”-attack (DOS-attack) går som namnet antyder ut på att på ett eller annat sätt hindra en tjänst från att fungera som det är tänkt [10]. Detta kan i p2p-fallet innebära att man bombarderar nätverket med data så att ingen kan ansluta, eller genom att man sprider så mycket falska filer att det blir omöjligt att hitta rätt.

Eftersom noderna i ett p2p-nätverk består av vanliga användare kan nätverken även attackeras genom att en användare ljuger om sina uppgifter. T.ex. skulle någon kunna

utge sig för att ha alla filer, och sedan skicka skräpdata som förstör nedladdningen för klienten. Detta blir särskilt effektivt i de fall där klienten laddar en stor fil från flera olika källor eftersom det då räcker med att ha skickat ganska lite data för att förstöra hela filen.

Fildelningsprogram kan även ha inbyggda buggar och andra säkerhetsproblem som kan utnyttjas i en DOS-attack. Tidigare versioner av Kazaa hade t.ex. en bugg som gjorde att klienten kunde krascha eller t.o.m. bli hackade och övertagna av någon annan [11].

När det gäller DOS-attacker med syfte att överbelasta nätverket får man skilja på rena p2p-nätverk och hybridnätverk [12].

## DOS-attacker mot rena p2p-nätverk

Ett rent p2p-nätverk kännetecknas av att det inte har någon server alls utan endast är uppbyggt med hjälp av noder. Attacker mot ett sådant nätverk måste alltså rikta sig mot själva noderna.

Ett nätverk som Gnutella som helt saknar servrar skulle man lätt kunna tro är immunt mot DOS-attacker, men så är inte fallet. En studie genomförd på Stanford University visar att Gnutella är mycket sårbart mot ”query”-attacker, d.v.s. att bombardera nätverket med förfrågningar och sökningar [13]. Alla förfrågningar propagerar runt på nätverket och hindrar noderna från att svara på de verkliga förfrågningarna. Enligt författarna skulle dock bättre belastningstekniker för noderna kunnat minska problemet och hindra dem från att bli överbelastade lika lätt.

En DOS-attack mot Gnutella skulle dock troligast bara lyckas ta ner delar av nätverket och inte hela, vilket fortfarande är en stor fördel mot serverbaserade nätverk i det avseendet.

Kazaa är ett annat rent p2p-nätverk som skiljer sig från Gnutella genom att det delar upp noderna i lövnoder och supernoder. En lövnod kopplar upp sig mot en eller flera supernoder som sedan håller reda på vilka filer som finns hos lövnoderna som hör till den. Supernoderna är sedan kopplade i ett nätverk sinsemellan. Detta förfarande gör att nätverket blir mer skalbart och snabbare för användarna men förenklar också en DOS-attack eftersom det nu räcker med att angripa supernoderna [14].

## DOS-attacker mot hybridnätverk

Ett hybridnätverk skiljer sig från ett rent p2p-nätverk genom att det på ett eller annat sätt utnyttjar en server. Fildelningsprogram som Napster och Direct Connect förlitar sig helt på en server som användarna får koppla upp sig mot för att sedan kunna genomföra sökningar och annat genom servern. En sådan modell är mycket sårbar för en DOS-attack eftersom det i princip bara är en dator som behöver attackeras.

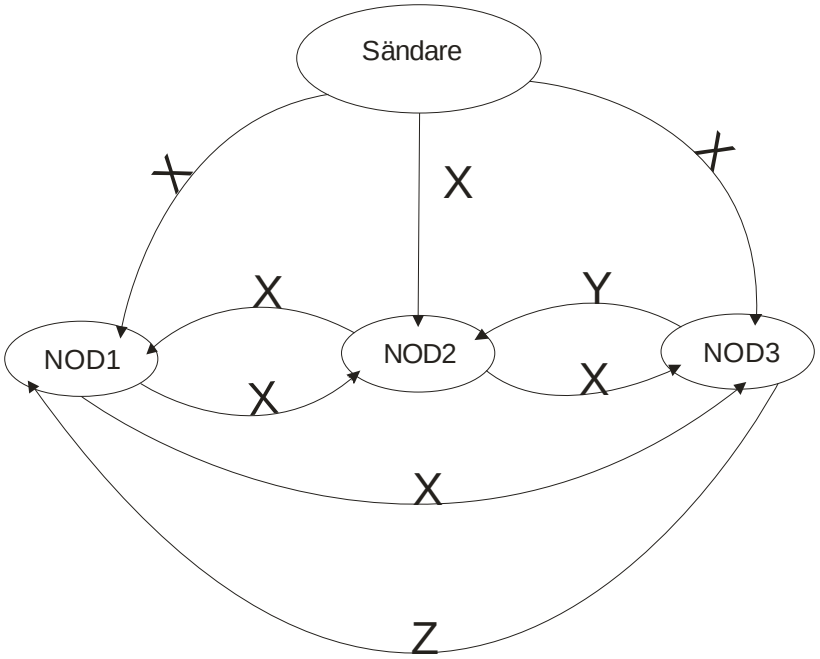
DOS-attacker mot hybridnätverk fungerar på samma sätt som de flesta DOS-attacker som genomförs på Internet. En vanlig metod är att låta ett flertal, vanligen hackade, datorer samarbeta genom att konstant försöka koppla upp sig mot servern. Det leder till att servern blir överbelastad och inte kan svara på fler anrop.

## *Bysantinska överenskommelsen*

*The byzantine generals problem* [9] behandlar en grupp bysantinska arméer som har omringat en stad. Varje armé har en general och för att lyckas med anfallet måste alla arméer anfalla samtidigt, eller helt sonika gå till reträtt. En del generaler föredrar reträtt och andra föredrar att anfalla, men vad som än bestäms måste alla vara med på det. Skulle anfallet genomföras utan alla arméer finns det nämligen stor risk för att det misslyckas. För att fatta beslut om vad som ska göras skickar därför alla generalerna meddelanden till varandra med vad de föredrar och sedan genomförs det förslag som fått majoritet. Problemet uppstår i det fall där en av generalerna är korrupt och skickar olika meddelanden till de övriga generalerna. Om det till exempel finns fem generaler där två vill anfalla och två vill gå till reträtt kan den femte, korrupta, generalen skicka olika meddelanden till de andra och på så vis få en del arméer att gå till anfall medan andra går till reträtt.

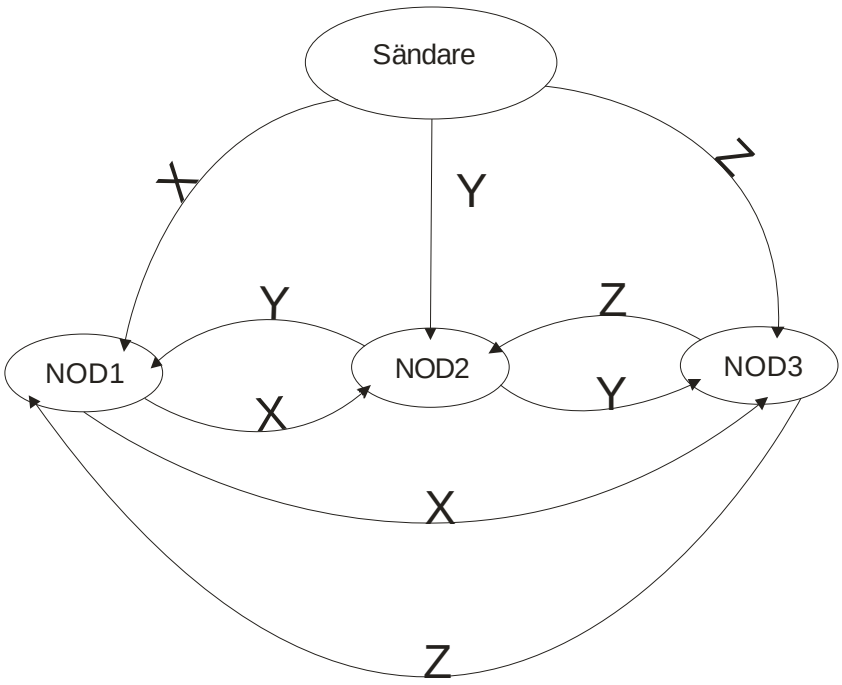
Lösningen på problemet är att låta generalerna vidarebefordra alla meddelanden de tar emot till de övriga generalerna. På så vis vet generalerna om vilka meddelanden som skickats till alla de andra generalerna och bedragaren kan avslöjas och beslutet fattas utan att denne har någon inverkan.

Det grundläggandet målet för den bysantinska överenskommelsen är att alla noder som inte är felaktiga ska kunna komma fram till ett enhetligt beslut och använda sig av samma värde. Detta kan man dra nytta av i p2p-nätverk där det är viktigt att utbyta information som måste vara enhetlig, t.ex. i distribuerade beräkningssystem. För att kunna upptäcka om en nod är felaktig krävs det minst 3f+1 noder totalt i systemet för varje felaktig nod f. Högst en tredjedel av noderna får alltså vara felaktiga [9].




*Figur 1.* Nod 3 är felaktig

I exemplet ovan är Nod 3 felaktig och vidarebefordrar olika data till de andra. De övriga noderna kan sedan genom majoritetsomröstning se att Nod 3 inte följer mönstret och därmed dra slutsatsen att den är felaktig och om så behövs utesluta den ur systemet.




*Figur 2.* Sändaren är felaktig

I exemplet som illustreras av *Figur 2* är sändaren felaktig och skickar olika information till de tre noderna. I det här fallet går det inte att avgöra vilket värde som är rätt, men eftersom alla noder använder sig av samma majoritetsomröstningsalgoritm kommer de att komma överens om att använda samma värde.

Det är dock värt att notera att den bysantinska överenskommelsen inte garanterar att ett värde är korrekt, utan bara att alla noder i systemet använder samma värde. För att garantera att man använder ett korrekt värde behöver man skapa ett system med redundans och olika typer av felkontroller för de olika värdena.

### *Övriga attacker*

Bland övriga attackformer kan nämnas att p2p-nätverk ger möjlighet för spam att skickas genom meddelanden. En spammare kan på ett snabbt och anonymt vis nå ut till många användare och genom att titta på vad folk delar ut för typ av filer kunna skicka reklam med riktat innehåll.

Som vi tidigare nämnt kan även malware spridas genom falska filer som användare laddar ner och kör i tron att det är någonting annat. Mycket av denna malware har till syfte att överskölja mottagaren med reklam och även kunna snappa upp personuppgifter. Originalversionen av Kazaa är t.ex. ökänt för att innehålla spyware som gör just detta.

Ytterligare en tänkbar attack är en klient som lägger till virus till alla filer som laddas ner från den. Viruset kan sedan ha till syfte att ytterligare attackera nätverket eller göra någonting helt annat.

## Sammanfattning

I rapporten har vi behandlat integritet och tillit i p2p-nätverk. Integriteten kan gälla både den data nätverket behandlar och de klienter som används för att ansluta till nätverket. Ett flertal problem finns, men användning av checksummor och hashning för att verifiera data och program motverkar de flesta. I slutändan är det dock tilliten som är den avgörande faktorn för om man bör använda en p2p-applikation.

Vidare har vi behandlat sekretess och användares personliga integritet när det gäller p2p-system. Sekretess kan vara en viktig del om datan som behandlas är känslig, och som i andra datorsystem så är kryptering ett bra sätt att skydda sekretessen. Användarnas personliga integritet kan också vara en känslig punkt i ett p2p-nätverk, och här finns lösningar som FreeNet som värnar om användarnas anonymitet.

Vi har även tittat på olika typer av attack mot p2p-nätverk och sett att det finns flera olika typer av hot. Förutom DOS-attacker riktade mot själva nätverken finns det attacker som försvagar nätverken på andra sätt. Dessa är oftast riktade mot fildelningsnätverk och handlar vanligen om sätt att förstöra filöverföringar eller att sprida dåliga filer.

P2p-nätverk är i grunden säkrare än vanliga typer av nätverk med en central server, men är trots detta inte helt oemottagliga för DOS-attacker. Ett rent p2p-nätverk kan

fortfarande utsättas för attacker mot själva nätverket som hindrar legitimerade användare från att kunna använda det. Rena nätverk som Kazaa som utnyttjar sig av s.k. supernoder är extra sårbart mot detta. Hybridnätverk med en central server är dock inte mer skyddat än något annat nätverk mot dessa typer av attacker och innebär således inte något extra skydd alls.

Den bysantinska överenskommelsen garanterar att alla noder kommer fram till samma resultat och kan även upptäcka noder som havererat och rapporterar felaktiga värden. Detta är främst till nytta inom distribuerade beräkningsnätverk.

## Slutsatser

Peer-to-peer-system är precis som andra datorsystem och –nätverk drabbade av säkerhetsproblem, och den utbredda användningen gör att det får en stor påverkan. Problemen är många och vitt skilda och hoten är både interna och externa, men de flesta problemen kan avhjälpas med vedertagna tekniker såsom kryptering för sekretess och hashning för integritet. Dessa lösningar börjar utnyttjas av applikationer på marknaden nu vilket tyder på att säkerheten har börjat tas på allvar även i p2p-nätverk och –applikationer.

Vidare märker vi att även andra aspekter av säkerhet börjar behandlas, speciellt genom specialiserade applikationer såsom nätverk. Nästa steg är att utnyttja flera av de lösningar vi har tagit upp för att ytterligare öka säkerheten i p2p-system.

Avslutningsvis kan vi även konstatera att om man är ute efter att skapa ett nätverk som är säkert och har hög upptid är p2p-nätverk ett intressant alternativ. Eftersom ett rent p2p-nätverk saknar centrala servrar är överbelastningsattacker svårare att genomföra men inte helt omöjliga, vilket vi har diskuterat i texten.

# Källförteckning


[1] *BitTorrent Frequently Asked Questions: I don't want you stealing my bandwidth! How can I stop it from uploading?* [www]. Hämtat från http://www.bittorrent.com/faq.myt#stopupload

[2] *Kazaa: Allegations of malware* [www]. Hämtat från http://en.wikipedia.org/wiki/Kazaa#Allegations_of_malware

[3] Piscitello, D., *Security and Peer-To-Peer Applications* [www]. Hämtat från http://cnscenter.future.co.kr/resource/hot-topic/p2p/2002-10-piscitello.pdf

[4] Kahney, L., *Cheaters Bow to Peer Pressure* [www]. Hämtat från http://www.wired.com/news/technology/0,1282,41838,00.html

[5] *BitTorrent Frequently Asked Questions: How do I know the download isn't corrupted?* [www]. Hämtat från http://www.bittorrent.com/faq.myt#corruptdl

[6] *Transport Layer Security* [www]. Hämtat från http://en.wikipedia.org/wiki/Secure_Sockets_Layer

[7] *The Free Network Project* [www]. Hämtat från http://freenet.sourceforge.net/

[8] McKean, C., Peer-*to-Peer Security and Intel's Peer-to-Peer Trusted Library* [www]. Hämtat från http://www.sans.org/rr/whitepapers/threats/468.php

[9] *Byzantine fault tolerance* [www]. Hämtat från http://en.wikipedia.org/wiki/Byzantine_fault_tolerance

[10] CERT® Coordination Center, *Denial of Service Attacks* [www]. Hämtat från http://www.cert.org/tech_tips/denial_of_service.html

[11] *Kazaa Sig2Dat Protocol Remote Integer Overflow and Denial Of Service by creating files in arbitrary locations* [www]. Hämtat från http://lists.grok.org.uk/pipermail/full-disclosure/2005-January/030999.html

[12] *Peer-to-Peer* [www]. Hämtat från http://en.wikipedia.org/wiki/Peer-to-peer

[13] Daswani, N., Garcia-Molina, H., *Query-Flood DoS Attacks in Gnutella* [www]. Hämtat från http://www.cs.vu.nl/~crispo/teaching/atcs/P2P/query-flood.pdf

[14] *Denial-of-Service Resilience in Peer-to-Peer File Sharing Systems* [www]. Hämtat från http://labos.epfl.ch/webdav/site/labos/shared/import/migration/Evil_P2P_DoS_attacks.pdf