FFI-rapport 2013/03053

Trafikkdatalekkasje gjennom bruk av sosiale medier

Lasse Øverlier

FFI Forsvarets forskningsinstitutt

Trafikkdatalekkasje gjennom bruk av sosiale medier

Lasse Øverlier

Forsvarets forskningsinstitutt (FFI)

2. september 2014

FFI-rapport 2013/03053

1242

P: ISBN 978-82-464-2414-9
E: ISBN 978-82-464-2415-6

Emneord

Soisale medier

Apper

Datalekkasje

Trafikkdataanalyse

Mobile enheter

Godkjent av

RonnyWindvik	Prosjektleder
Anders Eggen	Avdelingssjef

Sammendrag

Ukrypterte forbindelser forteller alt om hvem som kommuniserer og hva de utveksler av informasjon. Krypterte forbindelser skjuler innhold, men ikke hvem som kommuniserer. Mens anonymiserte forbindelser skjuler både innhold og de kommuniserende partene. Vi har tatt for oss bruk av tre av de mest ubredte sosiale tjenester i disse tre settingene på mobile enheter og undersøkt om de sosiale mediene lar seg identifisere. To av tjenestene kan trolig identifiseres i alle tre scenarioene, mens den tredje kun kunne identifiseres i ukrypterte forbindelser dersom man så bort fra DNS og IP-adresser.

English summary

Unencrypted connections reveals everything about who is communicating and what information they exchange. Encrypted connections hide the content but not who is communicating. But anonymized connections hide both content and the communicating entities. We have looked at three of the most popular social media services in all three settings on a mobile platform and examined whether the social platform is identifiable. Two of the services were likely identifiable in all three scenarios, but the third service could most likely only be identified in unencrypted networks if we ignore DNS and IP-addresses.

Innhold

1 Innledning

Med dagens ekstreme ekspansjon og sterkt økende bruk av sosiale medier blir det mer og mer viktig hvor mye informasjon man legger igjen gjennom interaksjonen med disse. Mange artikler og rapporter er skrevet[11, 14] om informasjonen brukerne legger inn i de sosiale mediene. Men det har vært lite fokus på om det er gjenkjennelig på andre måter ved at man er en bruker av sosiale medier. Dersom man ikke har tilgang til innhold i kommunikasjonen, men kun kan observere trafikk, lar det seg da gjennomføre å si at en bruker faktisk deltar i sosiale medier, og hvilke sosiale medier brukeren har interaksjon med?

Denne rapporten tar for seg trafikkdata til/fra en mobiltelefon som benytter et utvalg av sosiale medier. Vi vil i Kapittel 2 beskrive litt av den relevante bakgrunnen i temaet trafikkanalyse. Kapittel 3 vil beskrive eksperimentene som er laget samt en enkel vurdering av innsamlede data, og Kapittel 4 vil oppsummere.

2 Bakgrunn

Trafikkdataanalyse er noe helt annet enn innholdsanalyse. Dersom man kan ta for seg nivåer av informasjonstilgang i en "normal"[1] datakommunikasjon, vil man vi i vårt trafikkanalysescenario ønske å dele denne kommunikasjonen inn i tre kategorier:

- Vanlig kommunikasjon mellom to parter.
- Kryptert kommunikasjon mellom to parter.
- Anonymisert trafikk.

I vanlig kommunikasjon kan man for eksempel tenke seg bruk av Voice-over-IP (VoIP) eller vanlig web-surfing uten kryptering. Her er all trafikkinformasjon og alle data tilgjengelige i kommunikasjonskanalen. Her vil det lett avgjøres hvem brukeren kommuniserer med, samt innholdet i kommunikasjonen.

Ved kryptert kommunikasjon, for eksempel bruk av kryptert VoIP eller surfing i nettbanken, er kun innholdet i kommunikasjonen skjult i kommunikasjonskanalen. Det er ikke gjort forsøk på å skjule hvem som kommuniserer, og oftest heller ikke hva slags informasjon som utveksles (kryptert VoIP tilsvarer oftest tale-/videosamtale, surfing i nettbanken tilsvarer nettsider med bankinformasjon). Dette kan kategoriseres som metadata og er grunnlaget for store deler av diskusjonene rundt Datalagringsdirektivet i EU[9] og lagringen av telefoniinformasjon i USA[10, 12, 13].

Anonymisert trafikk kan forekomme for eksempel gjennom bruk av VPN og proxy-servere. Virtual Private Network (VPN) og proxy-noder er en måte å knytte seg opp i et annet nettverk

[1] Det er umulig å definere "normal", men i denne konteksten forsøker vi kun å dekke de mest vanlige benyttede måter for datakommunikasjon.

på. Her vil all trafikk (oftest) gå samlet gjennom den krypterte VPN/proxy-forbindelsen og blandes med all annen trafikk i dette nettverket før man kommuniserer med tjenester utenfor det interne nettverket. I tillegg blir all intern kommunikasjon i nettverket beskyttet på samme måte. Det finnes også anonymiseringsnettverk[8] og anonymiseringstjenester[1, 7] som i prinsippet fungerer på samme måte. Den største forskjellen ligger i hvor mye informasjon som VPN-leverandøren/proxy-eieren/anonymniseringstjenestene har mulighet til å samle om brukeren. Her *kan* metainformasjon være tilgjengelig hos leverandøren, men noen av tjenestene skjuler også all metainformasjon. Vi definerer derfor i vårt tilfelle anonymisert trafikk til de tilfellene hvor man også skjuler metainformasjonen.

Kort oppsummert er tilgjengeligheten av data vist i følgende tabell:

	Tilgjengelig informasjon:		
Trafikktype	**Innhold**	**Endepunkter**	**Metadata**
Vanlig trafikk	Ja	Ja	Ja
Kryptert trafikk	Nei	Ja	Av og til
Anonymisert trafikk	Nei	Nei	Nei

Vi har i en tidligere FFI-rapport[15] sett på lekkasje i kommunikasjonen fra mobiltelefonene til/fra produsent og tjenesteleverandør. Her ble det sett på innholdet i åpne og krypterte kommunikasjonskanaler for å finne mulig informasjonslekkasje om telefonen og/eller brukeren. Med dette som bakgrunn ønsket vi derfor å se på trafikkdata og undersøke hva slags informasjon en enkel vurdering av trafikkdata fra mobiltelefoner kan gi.

Vi har sett bort fra de åpenbare lekkasjene og informasjonen som DNS-oppslagene normalt gir oss. Dette kommer også frem senere i rapporten. Vi går heller ikke i dybden ved knytningen av IP-adresser til distinkte sosiale tjenester, ettersom dette også vil gi tilstrekkelig informasjon ved ukryptert bruk i dagens Internett. Ved å se bort fra DNS-oppslag og IP-adresser gjør vi problemstillingen om til en trafikkdataanalyse. Denne analysen ser på om det er mønster i selve kommunikasjonen som lekker informasjon om hvilke sosiale tjenester som benyttes. Dette blir derfor mer relevant i krypterte kommunikasjonskanaler som VPN, eller anonymisert trafikk.

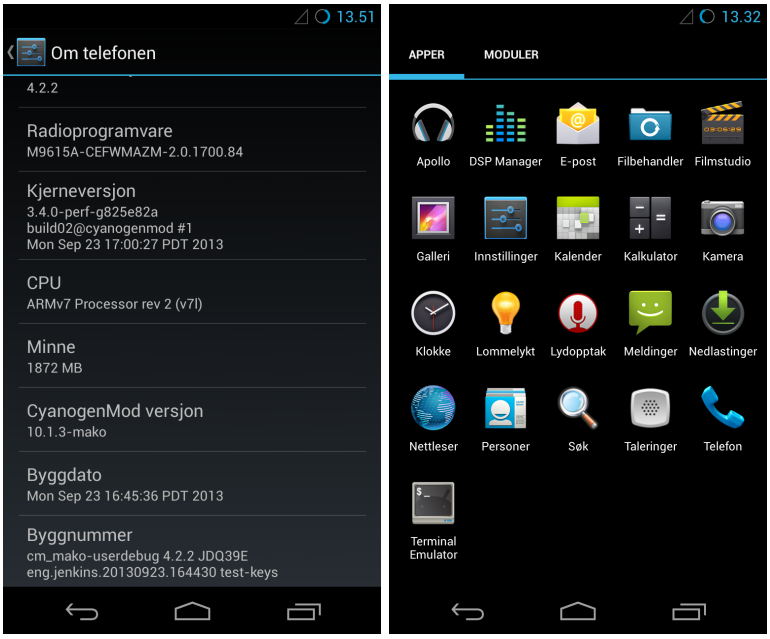
Vi har ønsket å ta utgangspunkt i de mest populære sosiale mediene, se på trafikkdata som genereres av disse appene på en vanlig smarttelefon, og se om disse kunne fortelle noe om hvilke sosiale medier som eventuelt ble benyttet.

3 Eksperimenter

Vi vil i dette kapittelet ta for oss oppsettet av mobile enheter, programvare og utstyr, samt vise hvordan eksperimentene ble gjennomført.

3.1 Oppsett

I forsøket ble det benyttet to telefoner av typen Nexus 4, med internt Android-navn “mako”. Disse er produsert av LG på oppdrag fra Google og kjører siste versjon av Android. Hovedgrunnen bak valg av disse telefonene ligger i at de er særdeles enkle å eksperimentere med, at man kan lage sine egne installasjoner av både Android-systemet og alle tjenester, samt at stort sett alle applikasjoner fungerer normalt på dem.



Figur 3.1 Telefoninformasjon ved en “ren Cyanogenmod”-installasjon.

Vi ønsket å fjerne det meste av støy fra telefonene og installerte derfor en versjon av Cyanogenmod[3] uten å ta med alle Google-appene som hele tiden “ringer hjem” til Google. Fordelen med Cyanogenmod er at den er minimalisert til å inneholde kun det nødvendige, samt mulighet for enkel endring av operativsystemet. Vi har tidligere tatt for oss analyse av trafikken som “ringte hjem” i nevnte FFI-rapport[15]. Mer detaljert informasjon om versjonsnummer og konfigurasjon er vist i figur 3.1.

Cyanogenmod versjon 10.1.3 kom også med svært få forhåndsinstallerte apper som vist til høyre i figur 3.1. Vi hadde også slått av automatiske oppdateringer av programmer og telefonens operativsystem, samt at vi kun installerte de programmene (“appene”) vi ønsket å ha med i eksperimentet.

Det finnes et utall av sosiale medier og kommunikasjonstjenester som man kan være en del av på Internett. For å holde eksperimentet forholdsvis enkelt har vi kun valgt ut tre av de mest populære[2] - Twitter, Facebook og Skype.

[2]Hva som er mest populært i dette ekstremt dynamiske og populære området forandrer seg fra år til år. Og det er derfor ganske sannsynlig at det er andre mer populære tjenester når dette leses.

Installasjonen av programmene ble gjennomført før trafikkdata ble observert, men brukerregistrering er gjort innenfor den aktuelle perioden. Vi samlet inn i overkant av førti timer med loggføring av trafikkdata hvor det bevisst ble gjennomført minimalt med brukerinteraksjon til de sosiale mediene. Grunnen til dette var at vi ønsket at det var standardinstallasjon av programmene som eventuelt skulle avsløre informasjon, ikke brukerens interaksjon med de sosiale mediene i seg selv. Vi ville altså undersøke om programmene avslører seg selv, eller om det ikke er mulig å skille installasjon eller bruk av ett spesielt program fra annen trafikk.

3.2 Hva kan lekke informasjon?

Uten brukerens interaksjon får man reletivt sett mindre informasjonsmengder enn når brukeren er aktiv. Det er bare observert trafikk, og eksperimentet har ikke påvirket trafikken på noen som helst måte[3].

Elementer som man kan se på i trafikkdata i såkalt "passiv modus" er typisk:

- DNS-navn/IP-adresser som benyttes
- Oppkoblingsdata og -frekvens
- Oppdateringsfrekvens
- Datamengde

Noe av dette skjer i klartekst, mens vi forventet at det meste av dataforbindelsene til tjenestene var kryptert og at man i beste fall kun kunne observere metainformasjon.

Frekvens på oppkoblinger kan være varierende eller den kan være fast for en eller flere av tjenestene. Dersom man kombinerer datamengde med frekvens kan man kanskje få en bedre indikator. I tillegg kan metainformasjon bidra ekstra ved for eksempel å koble IP-adresser til en spesifikk tjenesteleverandør, men dette er det som beskrevet tidligere ikke lagt vekt på her.

Vi vil forsøke å se om det finnes slike indikatorer for hver av tjenestene.

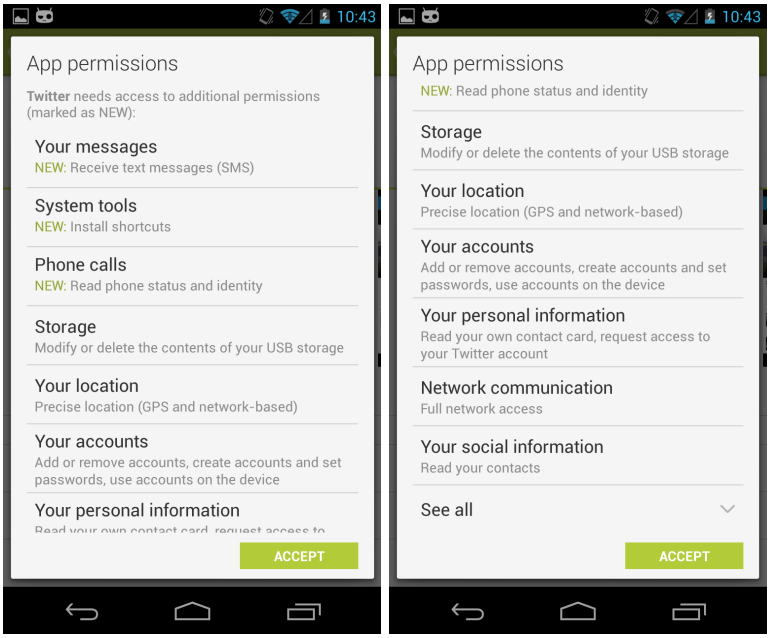
3.3 Twitter

Twitter er en interaksjonsplatform og et nettsamfunn hvor deltakerne kan registere en villkårlig identitet, lese innlegg fra andre og poste egne meninger, linker, meldinger på tilsvarende måte som i de fleste sosiale medier. Det noe spesielle med Twitter er en maksimallengde på meldinger på 140 tegn som begrenser informasjonsmengden i hvert innlegg/melding. Twitter er kanskje fra begynnelsen av sett på som "de gamles" sosiale plattform, men har i nyere tid vært svært aktivt i gryende undergrunnsbevegelser i undertrykkende regimer. Av denne grunnen blir Twitter ofte sett på som et problem i mindre demokratiske land.

Twitter er et relativt enkelt konsept og inneholder ikke mye data ettersom de kun formidler meldinger av begrenset størrelse og (i utgangspunktet) ingen bilder, o.l. Derfor ville vi forvente

[3] Som for eksempel gjennom Man-in-the-Middle angrep i [15]

en enkel og kompakt tjeneste med få servere involvert og bruk av kryptografi for å beskytte innlogging og publiseringer.



Figur 3.2 Twitters installasjonsrettigheter.

Hva Twitter krever av rettigheter er vist i figur 3.2. De ønsker blant annet tilgang til å kunne motta SMS, kontaktene til brukeren og lokasjonen til telefonen. Spesielt kontaktene er interessante for applikasjoner for sosiale nettverk ettersom det kan hjelpe dem med å koble sammen personer som allerede har en eller annen felles forbindelse. Hvorfor Twitter krever tilgang til å motta SMS gis det ingen begrunnelse på.

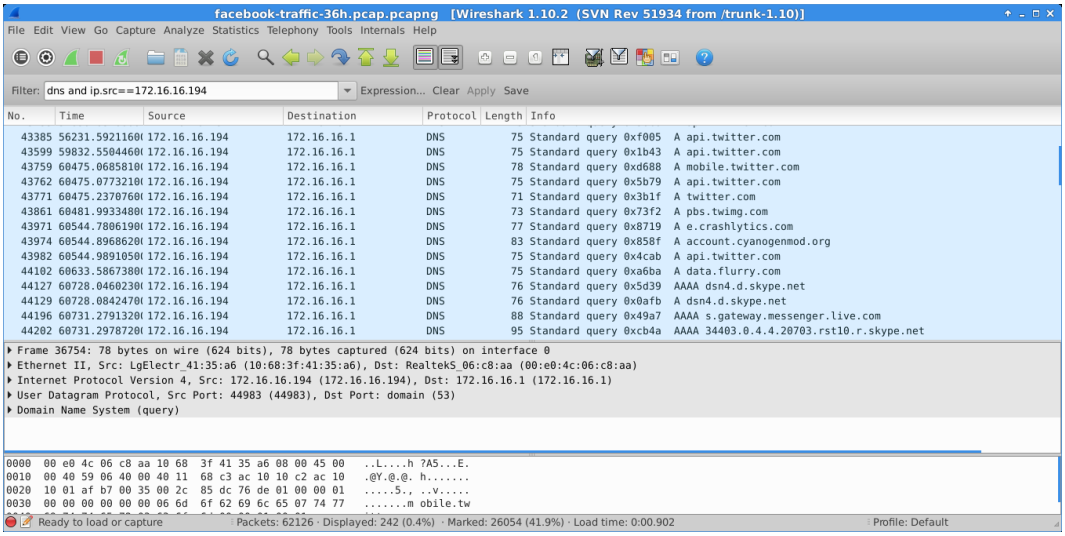
DNS og IP-adresser

Ettersom telefonene var relativt strippet for andre applikasjoner var det enkelt å sortere ut de DNS-forespørslene som var koblet mot Twitter. Figur 3.3 viser noen av DNS-forespørslene, og ved å sjekke svarene på disse (en liste av IP-adresser) ble det enkelt å sortere ut hvilke endepunkter som tilhørte Twitter.

Grunnen til dette er å kunne sortere ut trafikken som var relevant for denne sosiale tjenesten, for deretter å kunne gjøre analyse kun av trafikk knyttet mot denne tjenesten.

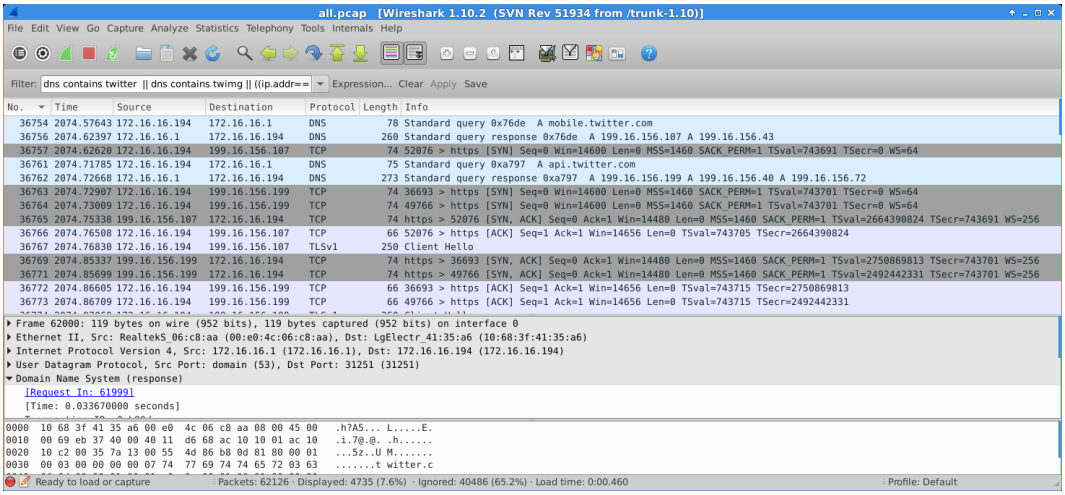
Wireshark er et verktøy for å se på nettverkstrafikk som gjør det mulig å lage filtre slik at man kun trekker ut det man er interessert i til enhver tid. Dessuten hjelper *Wireshark* til med å tolke data fra neste alle kjente protokoller. Vi har benyttet *Wireshark* til å hjelpe oss med å gjøre en vurdering av trafikken som vist i figur 3.3. For å søke etter informasjon relatert til Twitter ble følgende filter benyttet til å sortere ut all relevant trafikk:

```
dns contains twitter || dns contains twimg ||
```



Figur 3.3 DNS-bruk i Twitter.

```
((ip.addr==68.232.35.139 || ip.addr==199.96.57.7 ||
  ip.addr==199.16.156.0/24 || ip.addr==199.59.148.0/24 ||
  ip.addr==199.59.150.0/24))
```

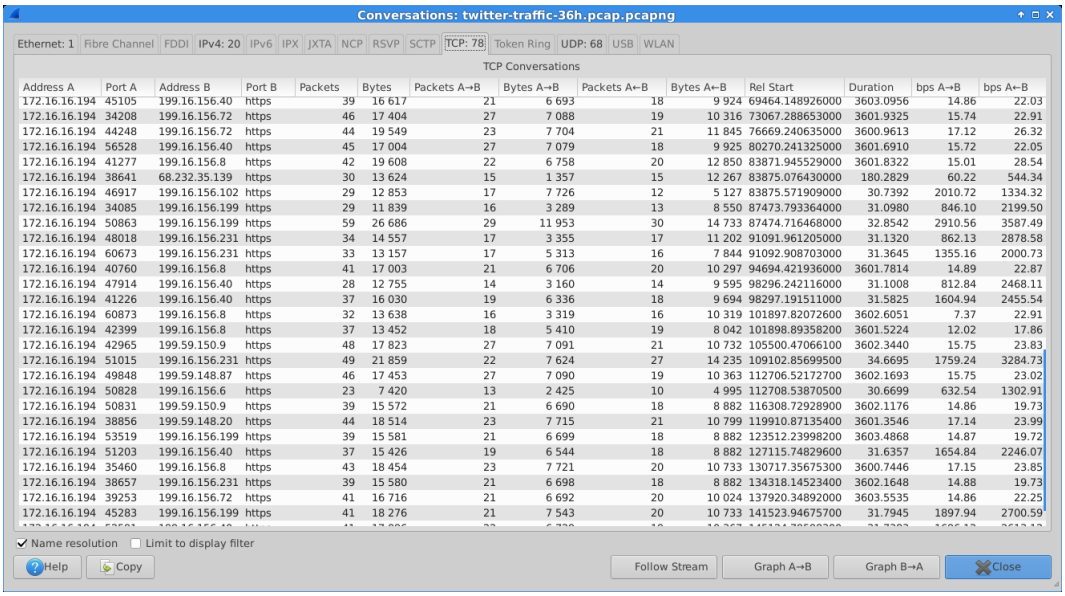


Figur 3.4 Twitter benytter alltid HTTPS.

Når vi tar en videre titt på trafikken til og fra Twitter-endepunktene så observerer vi i figur 3.4 at Twitter benytter kryptering på all kommunikasjon. Det benyttes vanlig SSL/TLS kryptering som har sine svakheter[15] og bugs[6], men som fortsatt kan regnes som sikker dersom den implementeres og brukes korrekt.

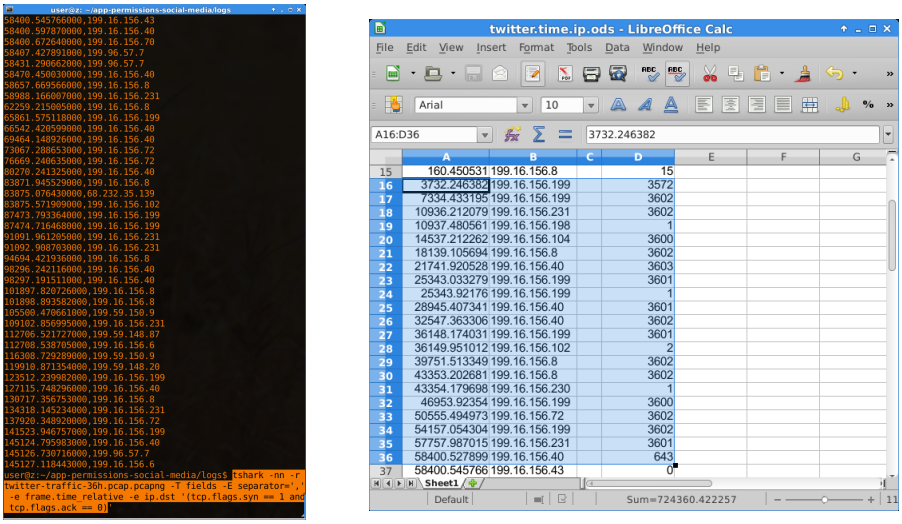
Det å gjennomføre at all kommunikasjon er kryptert har tidligere vært spesielt krevende og dyrt for de store aktørene på Internett og gjort kryptering til et mindre gjennomført sikkerhetstiltak enn ønskelig. Spesielt ettersom dette forlanger en del ekstra ressurser fra servere i flere nivåer. Men som nevnt tidligere - kryptering skjuler bare innhold, en god del metainformasjon om

trafikken er fortsatt tilgjengelig.



Figur 3.5 Trafikkmønster i Twitter.

Dersom vi ser litt grundigere på trafikkdata fra Twitter-interaksjonene, vist i figur 3.5, er det lett å se at det skjer noe periodisk. Vi ser blant annet at varigheten til en oppkobling har en tydelig overrepresentasjon av tidsvarighetene 3600 og 30 sekunder. Dermed kan man filtere ut hvor ofte nye oppkoblinger til tjenesten (på forskjellige servere) foregår, som vist i figur 3.6.



Figur 3.6 Skilt ut oppkoblingstidspunkter og tidsdifferanser mot Twitter-servere.

Figuren viser også at det er et tydelig gjenkjennelig mønster for oppkoblinger til Twitter-servere med eksakt en times mellomrom. Dette er et sett av servere som vi kan finne tilhørigheten til, for eksempel gjennom bruk av *traceroute*, i et nettverk kontrollert av domenet *twttr.com*[4] som

[4]Twitter benyttet også dette domenet i en 1. april spøk i 2013 som gikk ut på at alle Twitter-brukerne kunne benytte twttr.com dersom de skrev kompakt og uten konsonanter:)

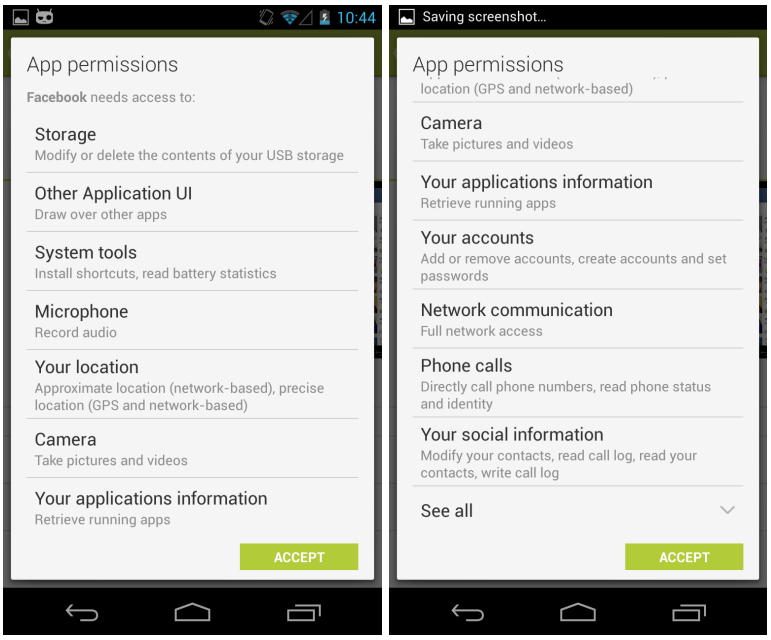
naturlig nok også tilhører Twitter Inc.

I figur 3.5 ser vi også på datamengden som overføres i hver retning. Kolonne seks, åtte og ti viser henholdsvis totalt overført data, data fra klient til server, og data fra server til klent. De aller fleste forbindelsene av en times varighet ligger på 13-19kb, de korte 30 sekunders forbindelsene noe mer variabelt 7-27kb med hovedvekt rundt 15kb. Andelen av trafikk til serveren ser ut til å ligge på rundt 15-35% av total trafikk - altså alltid mindre trafikk til serveren enn tilbake til smarttelefonen.

Applikasjonen har stort sett alltid én åpen forbindelse til en Twitter-server av gangen, men i perioder kan den av og til være uten forbindelse og noen ganger ha to parallelle oppkoblinger.

3.4 Facebook

Facebook, for tiden verdens største sosiale interaksjonsplatform, var et annet naturlig valg å se på. Her er det naturlig å anta at det er flere servere involvert, at det overføres mer data og kanskje at frekvensen på oppdateringene øker.

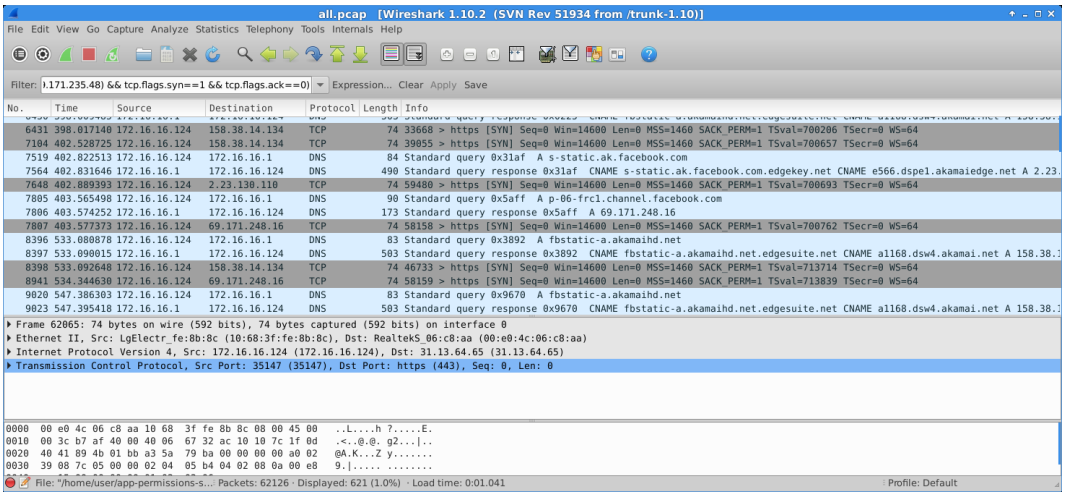


Figur 3.7 Facebook-app installasjonsrettigheter.

Først tok vi en titt på rettighetene som Facebook-appen krever. Vist i figur 3.7. Her kreves det, som hos Twitter-app, tilgang til nettverk, kontakter (lese **og skrive**), lokal lagring, og lokasjon. Men i tillegg vil Facebook ha tilgang til å bruke kamera og mikrofon, vise informasjon "oppå" andre programmer (dvs. benytte skjermen over et annet program), samt ringe og motta samtaler.

For å forsøke å finne aktuelle serveradresser må vi også her til med litt søk etter DNS-relatert informasjon. En titt gjennom logg forteller oss at det er et sett med domener som kan kobles mot Facebook-oppkoblingene i dette eksperimentet. Disse er facebook.com, fbcdn.net, fbexter-

nal*, fbstatic*.



Figur 3.8 Facebook-relatert trafikk.

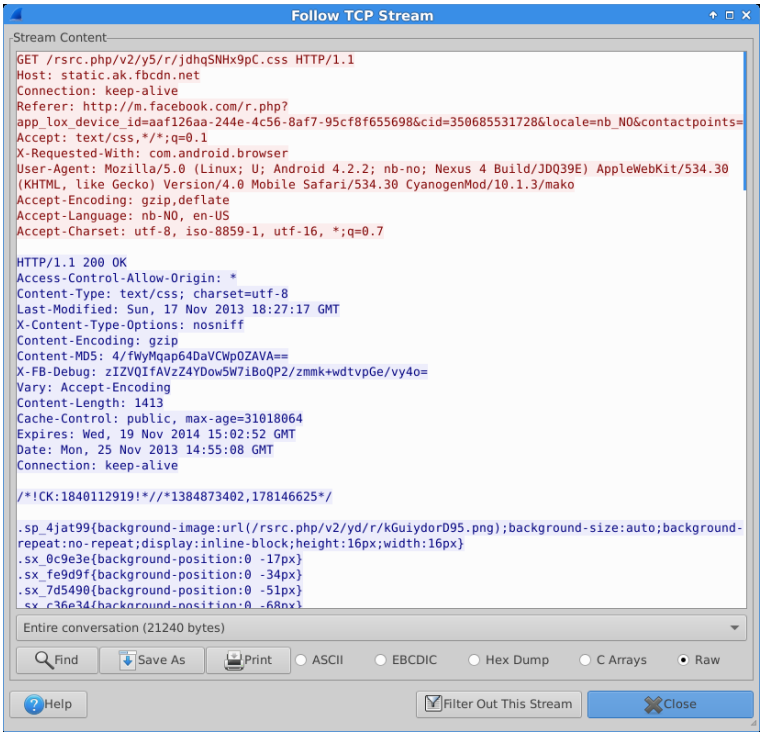
fbexternal, fbstatic, fbcdn.net er deler av DNS-navn på servere hos Akamai[2]. Disse serverene som kan sees på som midlertidige lagringsservere nærmere sluttbrukeren og applikasjonen. På denne måten sparer Facebook brukeren for tid ettersom serverne ligger lokalt i de fleste land, samt at de sparer Facebook-sentralene for trafikk ved å flytte statisk og semi-statisk informasjon som tekst, bilder og videoer vekk fra disse.

Uthentingen av trafikk-oppkoblinger mot Facebooks servere ble gjort med følgende filter:

```
dns contains facebook || dns contains fbcdn ||
dns contains fbexternal || dns contains fbstatic ||
((ip.addr== 173.252.110.29 || ip.addr==69.171.248.16 ||
  ip.addr==31.13.64.65 || ip.addr==173.252.103.16 ||
  ip.addr==173.252.110.29 || ip.addr==31.13.81.97 ||
  ip.addr==31.13.64.22 || ip.addr==31.13.81.97 ||
  ip.addr==31.13.81.113 || ip.addr==2.23.130.110 ||
  ip.addr==158.38.14.0/24 || ip.addr==148.123.13.61 ||
  ip.addr==69.171.235.48)
 && tcp.flags.syn==1 && tcp.flags.ack==0)
```

Her benytter vi siste linje til kun å trekke ut oppkoblingspakkene, dvs. første pakke som sendes mot Facebooks servere. Uten denne linjen ville vi fått ut all trafikk, men for oppkoblingsfrekvens trenger vi kun oppstarten. Bruk av dette filteret er vist i figur 3.8.

Etter en gjennomgang av informasjonen fra trafikkdata mot Facebook finner vi også ukryptert informasjon. Dette er spesielt sårbart gjennom mulighetene som åpner seg ved manipulasjon av overført data. En angriper kan endre data og for eksempel trigge sårbarheter i appen eller smarttelefonen gjennom disse ukrypterte forbindelsene.



Figur 3.9 Ukryptert innhold mot Facebook.

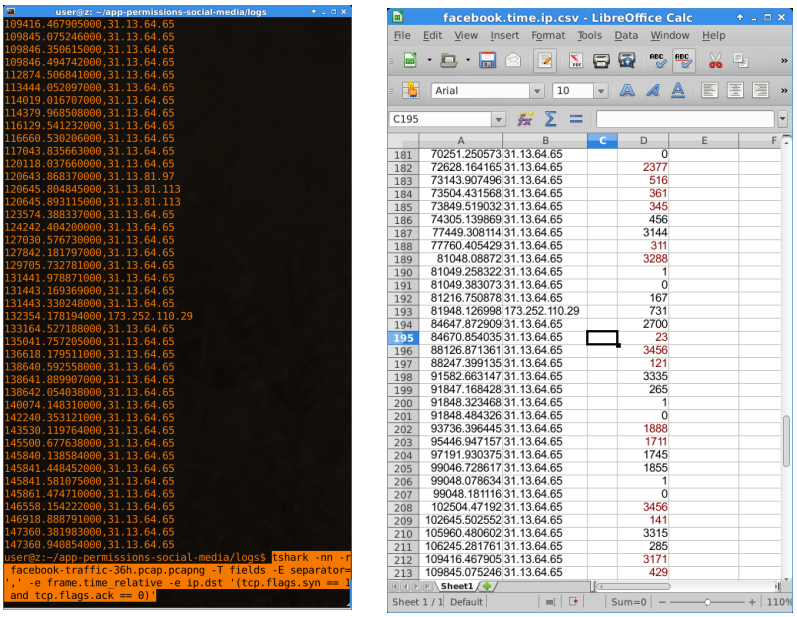
Ved å sortere ut oppkoblingstidspunkter og sortere oppkoblingene etter IP-adresse ved bruk av *tshark*[5] får vi ut oppkoblingsfrekvens som vist i figur 3.10. Det er også interessant å se på tallene merket med rødt og svart i regnearket til venstre. Vi ser at selv om Facebook har flere oppkoblinger med kortere opphold mellom oppkoblingene så har den et fast mønster med oppkoblinger for hver 3600 sekunder. Hver gruppe med røde eller svarte tall blir totalt tilnærmet 3600 sekunder, slik at første oppkobling i hver gruppe går presis med en times mellomrom. Disse oppkoblingene går i perioder ut til samme server.

Ved å se på forbindelsesstatistikken fra *Wireshark*, som vist i figur 3.11, kan vi også se oppkoblingsfrekvens, varighet av oppkoblinger, samt hvor mye data som ble overført i hver forbindelse.

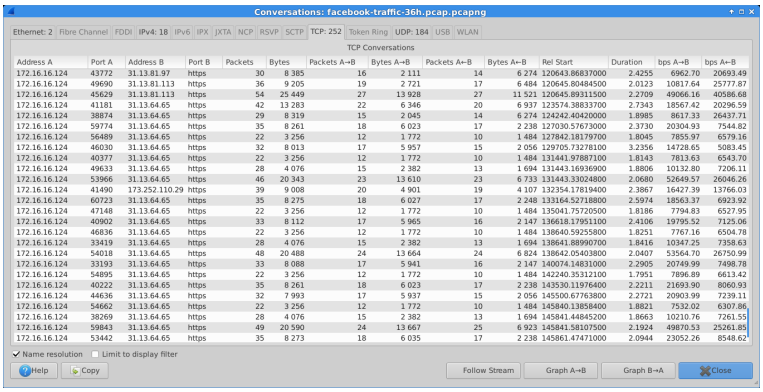
Av figuren ser vi at det er relativt faste størrelser på dataoverføringene. Sjette kolonne viser at det blir utvekslet ca. 3-4kb, 8kb eller 20kb med data i hver oppkobling mot Facebooks servere.

Noe som er mer uventet er retningen på data som utveksles. Vi skulle forvente at en oppdatering i sosiale medier bare sjekker på serveren om det er skjedd noe siden sist. Og dersom det er skjedd noe så henter applikasjonen ned alle forandringer. Kolonne åtte og ti viser data henholdsvis til og fra serveren. Ved de små overføringene er data til og fra fordelt omtrent likt. Ved "8k-oppkoblingene" går ca. 2kb tilbake til klienten, men 6kb går **til** serveren. Ved "20kb-oppkoblingene" går over 13kb **til** serveren og kun i overkant av 6kb tilbake. Ettersom dette er krypterte forbindelser har vi ikke forsøkt å se på hva slags informasjon dette er. Men det strider

[5] Kommandolinjeversjonen av Wireshark.



Figur 3.10 Oppkoblingstider og tshark-kommando for ekstrahering og sortering.



Figur 3.11 Oppkoblingsstatistikk for Facebook-forbindelsene.

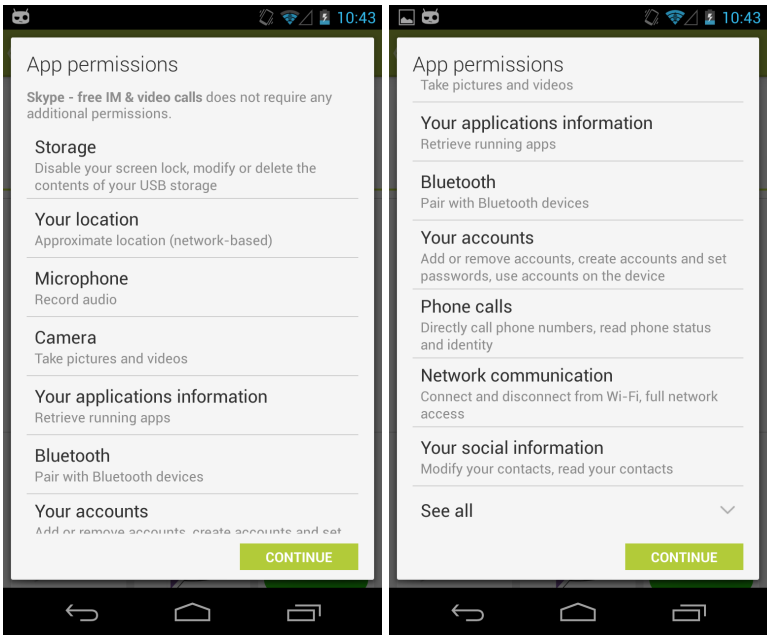
mot normal forventning at det går mer informasjon fra mobilen enn til mobilen ved stort sett alle typer automatiske oppdateringer.

Vi observerer også i kolonne elleve at "20kb-oppkoblingene" forekommer med ganske nøyaktig to timers (7200s) mellomrom.

3.5 Skype

Skype har i mange år vært verdens største gratis telefonisystem, med støtte for både en-til-en og mange-til-mange samtaler, og videosamtaler og -konferanser. Skype benytter en proprietær protokoll som mange har forsøkt å reversere[5]. Det er fortsatt ikke publisert noen komplett forklaring på hvordan protokollen er bygget opp. Nettverksstrukturen er ukjent men er bygget rundt et peer-to-peer prinsipp. Det er kjent at Skype benytter seg av såkalte supernoder med høy båndbredde som koblingspunkter for samtaler mellom brukerne. Disse supernodene kunne for

en tid tilbake være vanlige brukere, men ser nå ut til å være noder som Skype kontrollerer.



Figur 3.12 Rettigheter ved bruk av Skype-app.

Skype benytter også standard kryptografiske algoritmer som 256-bit AES og RSA, men det er fortsatt ukjent hvordan nøkkelgenerasjon forekommer slik at protokollen ikke lenger regnes som helt sikker[4] selv om krypteringen er krafitg nok mot innsyn fra ukjent tredjepart.

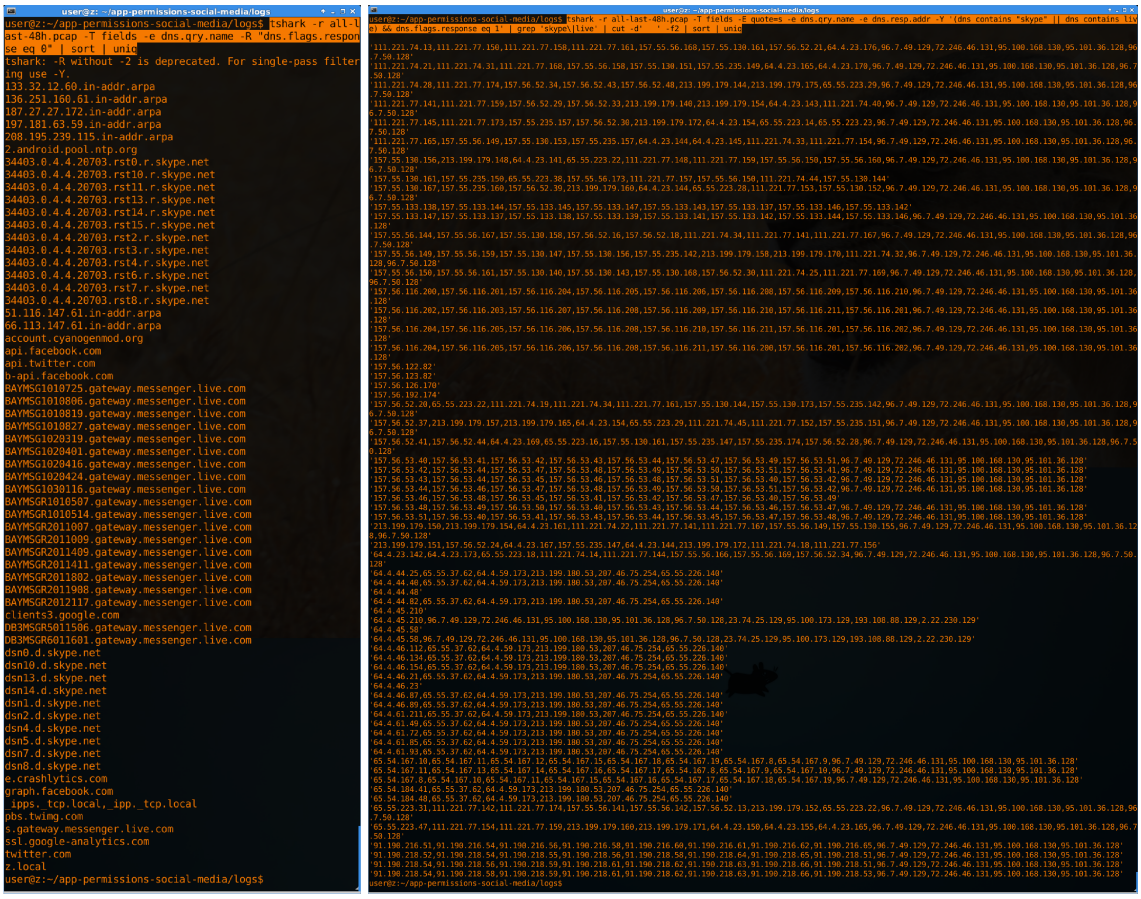
Figur 3.12 viser at installasjonsrettighetene Skype-appen ber om er tilnærmet de samme som Facebook-appen. I tillegg kommer også muligheten til å benytte Bluetooth. Skype har kanskje mer grunnlag for tilgang til mikrofon og å kunne ringe enn Facebook, men dette viser kanskje mer Facebooks ambisjoner og tenkte utvikling?

Ved å lete i loggene etter Skype-oppkoblinger får man et helt annet bilde enn ved bruk av Twitter og Facebook. Figur 3.13 viser DNS-oppslagene i en logg-periode og det er tydelig at Skype-oppslagene dominerer, spesielt dersom man regner inn **.live.com*-oppslagene som er til Skype-gatewayene.

Høyre del av figuren viser hvilke IP-adresser vi får ut fra DNS-oppslagene som er relatert til Skype-applikasjonen.

Dersom vi ønsker å få ut alle unike IP-adresser kjører vi dette videre gjennom et filter:

```
tshark -r all-last-48h.pcap -T fields -E quote=s \
  -e dns.qry.name -e dns.resp.addr \
  -Y '(dns contains "skype" || dns contains live) \
       && dns.flags.response eq 1' | grep 'skype\|live' | \
          cut -d'    ' -f2 | sort | uniq | cut -d "'" -f2 | \
```



Figur 3.13 Skype DNS bruk og resulterende IP-adresser.

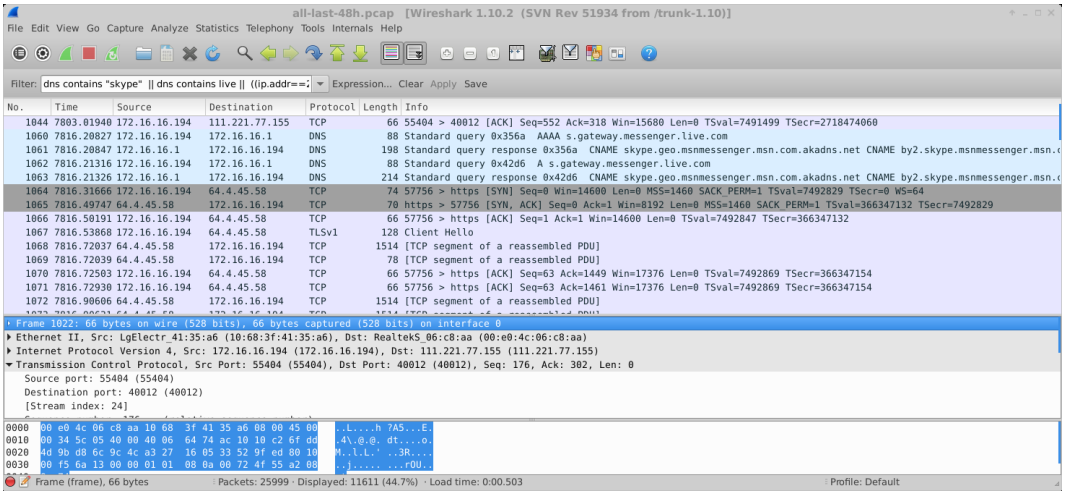
```
sed 's/,/\r\n/g' | sort | uniq
```

og vi er dermed nede i 242 unike IP-adresser involvert bare i vår korte loggeperiode.

For å generere et filter fra disse 242 adressene måtte vi generalisere litt og lage noen /24 subnett hvor vi antok Skype kontrollerte hele subnettet. Ble det dermed enkelt og oversiktlig? Ikke helt, men med et filter som vist under fikk vi frem all Skype-trafikken i Wireshark.

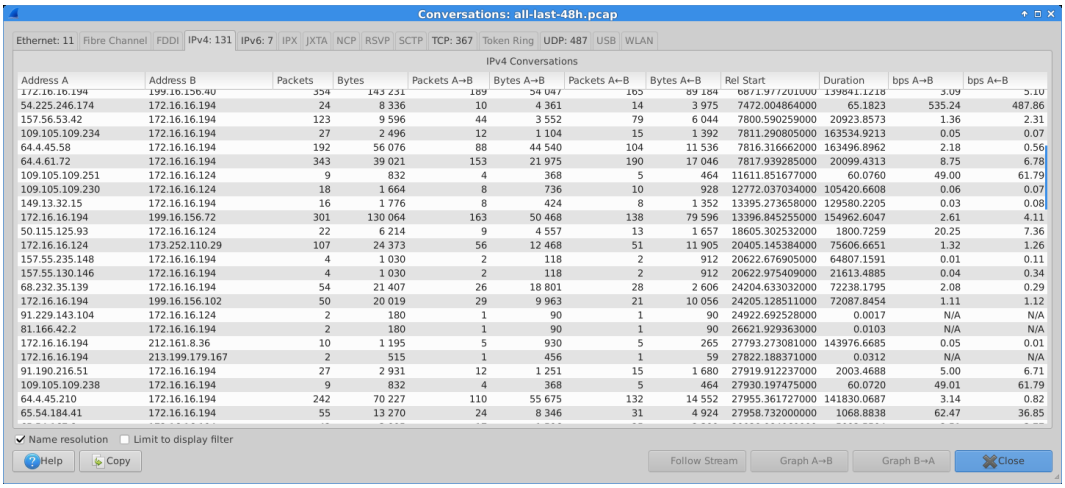
```
dns contains \skype"  || dns contains live ||
(ip.addr==2.22.230.129 || ip.addr==23.74.25.129 ||
 ip.addr==64.4.23.0/24 || ip.addr==64.4.44.0/18 ||
 ip.addr==65.54.167.0/24 || ip.addr==65.54.184.0/24 ||
 ip.addr==65.55.223.0/24 || ip.addr==65.55.226.140 ||
 ip.addr==65.55.37.62 || ip.addr==72.246.46.131 ||
 ip.addr==91.190.216.0/24 || ip.addr==91.190.218.0/24 ||
 ip.addr==95.100.168.130  || ip.addr==95.100.173.129  ||
 ip.addr==95.101.36.128  || ip.addr==96.7.49.129  ||
 ip.addr==96.7.50.128  || ip.addr==111.221.74.0/24  ||
```

```
ip.addr==111.221.77.0/24 || ip.addr==157.55.130.0/24 ||
ip.addr==157.55.133.0/24 || ip.addr==157.55.235.0/24 ||
ip.addr==157.55.56.0/24 || ip.addr==157.56.116.0/24 ||
ip.addr==157.56.122.82 || ip.addr==157.56.123.82 ||
ip.addr==157.56.126.170 || ip.addr==157.56.192.174 ||
ip.addr==157.56.52.0/23  || ip.addr==193.108.88.129 ||
ip.addr==207.46.75.254 || ip.addr==213.199.179.0/24 ||
ip.addr==213.199.180.53)
```



Figur 3.14 Skype ønsker også å kunne benytte IPv6 dersom den er tilgjengelig.

Da kunne vi observere at Skype også forsøker å kommunisere via IPv6 som vist i andre linje i datatrafikken i figur 3.14.



Figur 3.15 Skype trafikkdata i Wireshark.

I figur 3.15 viser vi litt av trafikkdata som er Skype-relatert. Som vi kan se i kolonnen for varighet, tredje kolonne fra høyre, er det ingen faste eller gjentakende lengder. Men mange

av forbindelsene holdes åpne i lang tid, gjerne et døgn eller flere dager. Skype har hele tiden mange åpne kommunikasjonskanaler i parallell og rett før den tar ned en kommunikasjonskanal kan den sette opp et par nye mot andre servere. Skype kan heller ikke observeres fra noen åpenbare mønstre i verken timing eller datamengde, men det er tydelig at det går en god del datatrafikk uten at det er noen interaksjon og aktivitet fra brukerens side.

4 Konklusjon

Vi har i denne rapporten sett kort på hvilke egenskaper som er mer og mindre tydelige gjennom trafikkdata for noen utvalgte sosiale medier.

Tabellen under viser elementer som er undersøkt i eksperimentene og hvordan de forskjellige applikasjonene har oppført seg.

	Applikasjon		
	Twitter	**Facebook**	**Skype**
Ukryptert trafikk	Nei	Noe	Nei
Periodisk oppkobling	Ja	Ja	Ikke funnet
Periodisk lik datamengde	Ja	Ja	Ikke funnet
Identifiserbare oppkoblingspunkter	Ja	Ja	Ja, men mange

Det er ikke mulig å komme til innhold i noen av scenariene, men på grunn av at Facebook ikke benytter kryptering på all kommunikasjon blir protokollen mer sårbar for angrep inntil dette er fikset.

Tabellen under viser om det er mulig å gjenkjenne bruk av applikasjonen gjennom vanlig trafikk, for eksempel i åpne trådløse nettverk, i krypterte forbindelser som VPN-tunneler, eller i anonymisert trafikk som for eksempel ved bruk av *Tor*.

	Applikasjon		
Identifiseres i:	**Twitter**	**Facebook**	**Skype**
Ukryptert nettverk	Ja	Ja	Ja
VPN/Kryptert trafikk	Ja	Ja	Trolig ikke
Anonymisert trafikk	Trolig	Trolig	Trolig ikke

I ukrypterte nettverk vil man kunne identifisere alle applikasjoner stort sett på grunnlag av DNS og IP-adresser, og dette var kjent fra tidligere.

På grunn av den tydelige periodiseringen og de gjentakende mengdene av data mener vi at bruk av Twitter og Facebook er identifiserbar i krypterte kommunikasjonskanaler og anonymiseringsnettverk. Skype derimot er det lite trolig at man kan gjenkjenne i verken krypterte forbindelser (uten bruk av DNS eller ip-adresser) eller i anonymiserte forbindelser ettersom de benytter svært lange perioder av ikke-deterministisk lengde og variabel mengde data i forbindelsene.

Referanser

[1] Anonymizer - Online Privacy and Personal VPN Software. https://anonymizer.com/. [Online; hentet 15. juni 2014].

[2] Cloud Services, Enterprise, Mobile, Security Solutions — Akamai. http://www.akamai.com/. [Online; hentet 15. juni 2014].

[3] CyanogenMod — Android Community Operating System. http://cyanogenmod.org/. [Online; hentet 15. juni 2014].

[4] Russian security services were given the opportunity to listen to Skype. http://www.vedomosti.ru/politics/news/10030771/skype_proslushivayu.

[5] Skype Reverse Engineering : The (long) journey ;).. — oKLabs. http://www.oklabs.net/skype-reverse-engineering-the-long-journey/. [Online; hentet 15. juni 2014].

[6] The Heartbleed Bug, CVE-2014-0160. http://heartbleed.com/. [Online; hentet 15. juni 2014].

[7] VPN – Secure Private Virtual Network Service. https://www.hidemyass.com/. [Online; hentet 15. juni 2014].

[8] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[9] European Parliament and Council. Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. *Official Journal of the European Union*, Mars 2006. [Online; hentet 15. juni 2014].

[10] Glenn Greenwald. Edward Snowden: the whistleblower behind the NSA surveillance revelations. http://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance. [Online; hentet 15. juni 2014].

[11] P. Jagtap, A. Joshi, T. Finin, and L. Zavala. Preserving privacy in context-aware systems. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 149–153, Sept 2011.

[12] S. Landau. Making sense from snowden: What's significant in the nsa surveillance revelations. *Security Privacy, IEEE*, 11(4):54–63, July 2013.

[13] S. Landau. Highlights from making sense of snowden, part ii: What's significant in the nsa revelations. *Security Privacy, IEEE*, 12(1):62–64, Jan 2014.

[14] Nan Li and Guanling Chen. Sharing location in online social networks. *Network, IEEE*, 24(5):20–25, September 2010.

[15] Lasse Øverlier. Data leakage from android smartphones. *FFI Rapport 2012/00275*, Juni 2012.