Bachelorscriptie
Informatiekunde

Radboud Universiteit

Smartphone-privacy

Auteur:
Ben Siebert
s0839671

Inhoudelijk begeleider:
Prof.dr. M.C.J.D. van Eekelen
marko@cs.ru.nl

Tweede lezer:
Fabian van den Broek, MSc
f.vandenbroek@cs.ru.nl

1 juli 2013

Samenvatting

Sinds de komst van de iPhone zijn smartphones razend populair geworden. Door de vele mogelijkheden van smartphones, zoals het maken van foto's en video's, chatten en het gebruik van GPS is de smartphone een verzamelplaats van persoonlijke gegevens geworden. Door een continue verbinding met het internet en het feit dat er partijen zijn die deze informatie graag willen hebben zou de smartphone een middel kunnen zijn om persoonlijke gegevens te bemachtigen.

In deze scriptie behandel ik de vraag welke gegevens verschillende apps versturen en waar naartoe. Om te kunnen zien welke gegevens apps versturen heb ik een opstelling gemaakt die man-in-the-middle-aanvallen uitvoert op apps zodat het netwerkverkeer van die apps inzichtelijk wordt.

Met de onderzoeksopstelling heb ik het netwerkverkeer van dertig apps opgevangen en geanalyseerd. Uit de analyse blijkt dat bijna alle geteste apps in meer of mindere mate privacygevoelige informatie versturen. De verzonden gegevens liepen uiteen van informatie over de smartphone tot informatie over de gebruiker daarvan. De gegevens werden zowel naar de makers van een app gestuurd, als naar derden zoals advertentiebedrijven. Het lijkt er dan ook op dat apps vaak ingezet worden voor het vergaren van gebruikersgegevens. Er is dan vaak ook geen duidelijk verband tussen de verzonden gegeven en de functionaliteit van de app.

Inhoudsopgave

Hoofdstuk 1

Inleiding

1.1 Opkomst smartphones

Smartphones zijn apparaten die primair een telefoon zijn maar nog veel andere functies hebben, zoals internet en muziek afspelen. Dit in tegenstelling tot 'dumbphones' waarmee je vrijwel alleen maar kunt telefoneren. Smartphones kenmerken zich door het hebben van een uitgebreid besturingssysteem en veel extra hardware-elementen zoals een camera, een touchscreen en een GPS-antenne.

Voordat smartphones populair werden onder het grote publiek gebruikten de meeste mensen reguliere mobiele telefoons zonder al te veel poespas. Daar kwam verandering in toen Apple de iPhone in 2007 introduceerde.[1] Vóór de introductie van de iPhone waren er ook al mobiele apparaten die meer konden dan een standaard mobiele telefoon, maar deze werden voornamelijk zakelijk gebruikt voor met name de agenda- en e-mailfuncties.

Met de iPhone maakte Apple de combinatie van telefonie, internet en muziek, wat leidde tot een succes. In 2008 voegde Apple de App-Store toe aan het besturingssysteem van de iPhone. Met de komst van de App-Store konden ontwikkelaars van software gratis of tegen betaling software aanbieden aan iPhone-gebruikers. Sindsdien is er voor bijna ieder denkbaar scenario een applicatie of kortweg 'app' beschikbaar, zoveel zelfs dat Apple een reclamecampagne met de slogan "there's an app for that"begon.[2]

Later in 2008 kwam Google met zijn besturingssysteem voor smartphones genaamd 'Android'.[3] In tegenstelling tot iPhone is Android niet gebonden aan één type toestel maar maken verschillende fabrikanten zoals HTC, LG en Samsung zogenaamde 'Androidtelefoons'. Inhoudelijk kunnen de verschillende typen smartphones nagenoeg hetzelfde, het verschil zit hem met name in details zoals het uiterlijk van het toestel of van de software. Android

[1] http://www.macworld.com/article/1054769/iphone.html
[2] http://www.youtube.com/watch?v=szrsfeyLzyg
[3] http://web.archive.org/web/20110712230204/http://www.htc.com/www/press.aspx?id=66338&lang=1033

en iPhone hebben op de dag van vandaag het grootste deel van de markt in handen, respectievelijk 60% en 20% van de Nederlandse smartphonemarkt.[4] Doordat beide een significant marktaandeel hebben zijn veel apps voor zowel Android als iPhone beschikbaar.

1.2 Privacyrisico

Omdat men veel persoonlijke dingen doet op een smartphone, zoals chatten en fotograferen, is de smartphone een verzamelplaats van persoonlijke gegevens geworden. De gemiddelde smartphone bevat onder andere belhistorie, sms-historie, internethistorie, foto's, video's en contactpersonen. Naast de opgeslagen gegevens bevatten smartphones hardware die real-time informatie zouden kunnen vastleggen. Denk hierbij aan live opnamen door de camera of microfoon, maar ook aan GPS- en telefoonmastgegevens. Ook bevatten smartphones veel identificatienummers die niet per se aan een persoon gekoppeld zijn, maar waarmee de smartphone zelf wel gevolgd zou kunnen worden.

Dat het voor bedrijven interessant is om persoonsgegevens te verzamelen blijkt uit de verzameldrift van gegevens door websites. Veel websites plaatsen zogenaamde 'tracking-cookies'.[5] Cookies zijn kleine tekstbestanden met daarin gebruikersvoorkeuren die een website kan uitlezen. Tracking-cookies zijn een speciaal soort cookies met gegevens waardoor gebruikers uniek identificeerbaar zijn en daarmee op het internet gevolgd kunnen worden. Omdat het volgen vaak zonder medeweten van de gebruiker gebeurt heeft de Nederlandse overheid wetgeving bedacht om dit in te perken en gebruikers beter op de hoogte te stellen.[6]

Google is naast ontwikkelaar van Android voornamelijk een grote speler op de advertentiemarkt en behaalde daarmee in 2012 nog 95% van zijn omzet.[7] Ook Apple houdt zich bezig met de advertentiemarkt en is in het verleden in Californië aangeklaagd voor het delen van persoonsgegevens met adverteerders.[8]

Als parallel met websites is het niet ondenkbaar dat er ook bedrijven bezig zijn met het verkrijgen van privacygevoelige informatie via smartphones. Een mogelijkheid hiervoor kan zijn dat apps gegevens van de gebruiker sturen naar de makers van de app, of misschien zelfs naar derden. Doordat de

[4] http://www.telecompaper.com/nieuws/android-groeit-tot-60-marktaandeel-in-nederland–925992

[5] http://www.consumentenbond.nl/test/elektronica-communicatie/veilig-online/privacy-op-internet/extra/wat-zijn-cookies

[6] http://www.rijksoverheid.nl/onderwerpen/ict/veilig-online-en-e-privacy/internetbezoek-volgen-met-cookies

[7] http://investor.google.com/financial/tables.html

[8] http://tweakers.net/nieuws/71625/apple-aangeklaagd-om-delen-persoonsgegevens-met-adverteerders.html

meeste smartphones permanent met het internet verbonden zijn zou dit op ieder moment dat de app actief is kunnen gebeuren. Het is voor de gebruiker ook niet altijd duidelijk wat er verzonden wordt omdat er geen visuele feedback hoeft plaats te vinden. Om gebruikers te beschermen tegen het zomaar verkrijgen van persoonsgegevens door apps hebben zowel Android als iPhone machtigingenschermen. Dit zijn overzichten van rechten die de app zegt nodig te hebben. Zo moeten ontwikkelaars van Androidapplicaties expliciet aangeven welke rechten hun app nodig heeft. Vervolgens wordt bij installatie een scherm met een verzoek tot machtiging getoond.

Figuur 1.1: Machtigingscherm in Android.

Helaas is het machtigingenscherm geen waterdichte methode om de privacy van de gebruiker te waarborgen omdat veel gebruikers niet eens weten wat de functie van het machtigingenscherm is. Daarnaast nemen veel gebruikers het bewust niet in acht (Felt e.a. 2012) of is het onduidelijk wat de verschillende soorten rechten inhouden die apps vragen. Een andere probleem met de machtigingenschermen is dat het alleen toegang vraagt tot hardware of informatie, maar het geen intentie geeft wat de app ermee gaat doen. Hierdoor kan de app een ogenschijnlijk goede reden hebben om de rechten te vragen, maar kunnen die rechten daarnaast misbruikt worden om bijvoorbeeld informatie door te sturen naar derden.

1.3 Onderzoeksvraag

Het doel van deze scriptie is om een inzicht te krijgen in wat voor gegevens verscheidene apps zoal versturen en waarheen ze die gegevens dan sturen. Voor dat doeleinde stel ik de volgende onderzoeksvragen:

1. Welke gegevens versturen apps?

2. Waar worden die gegevens naartoe gestuurd?

Om de onderzoeksvragen te beantwoorden zal ik een methode opzetten om het netwerkverkeer van een smartphone inzichtelijk te krijgen. Ik zet uiteen welke afwegingen gemaakt moeten worden om het onderscheppen mogelijk te maken en leid hieruit een proefopstelling af. Als laatste geef ik een weergave van het resultaat behaald met de proefopstelling en de analyse van de gegevens.

1.4 Gerelateerd werk

Op het gebied van privacy met betrekking tot mobiele applicaties is al het een en ander aan onderzoek gedaan. Toch denk ik dat mijn onderzoek waardevol is omdat het toch net een andere insteek heeft, of voorgaand onderzoek aanvult.

Eric Smith onderzocht het versturen van Unique Device IDentifiers ofwel UDIDs. Hij kwam tot de conclusie dat 68% van de apps UDID's versturen naar externe servers (Smith 2010). Dit brengt risico's met zich mee omdat een smartphone vaak persoonlijk is, en een UDID dus bijna 1-op-1 staat voor een gebruiker. De methode van Smith was om een smartphone via Wi-Fi te laten verbinden met een accesss point en vervolgens de access point het netwerkverkeer te laten opslaan. Versleutelde verbindingen werden in dit onderzoek niet meegenomen.

The Wall Street Journal plaatste een reeks artikelen onder de naam "What they know". Hierin onderzochten zij het spionagegedrag van marketeers op internetgebruikers. In december 2010 verscheen het artikel "What They Know - Mobile"waarbij apps getest werden op het versturen van privacygevoelige informatie.[9] Het resultaat was een uitgebreide weergave van welk type gegevens bepaalde applicaties rondstuurden en naar welke bedrijven. Bij het artikel is ook een korte uitleg over de methode, namelijk dat een Wi-Fi-verbinding gebruikt werd om het netwerkverkeer te isoleren en op te slaan en dat een tool genaamd 'Mallory' gebruikt werd voor het ontsleutelen.

Yildirim (2012) beschreef een methode om netwerkverkeer te onderscheppen door middel van Wi-Fi en hoe versleutelde verbindingen (https) inzichtelijk gemaakt konden worden. Focus van zijn onderzoek lag met name bij het juridische aspect van smartphone-privacy waardoor de hoeveelheid geteste applicaties klein was.

[9] http://blogs.wsj.com/wtk-mobile/

Hoofdstuk 2

Meten van netwerkverkeer

2.1 Opvangen verkeer

Voordat het netwerkverkeer van een smartphone geanalyseerd kan worden zal het eerst opgevangen moeten worden. In dit hoofdstuk zet ik verschillende methoden voor het opvangen van netwerkverkeer van een smartphone uiteen.

2.1.1 Netwerkdump op telefoon

Op Android is het mogelijk om het netwerkverkeer via een applicatie weg te laten schrijven naar een bestand. Programma's als tcpdump of tPacketCapture kunnen het netwerkverkeer wegschrijven naar 'packet capture'-formaat (pcap) dat vervolgens op een pc uitgelezen kan worden door een netwerkanalyseprogramma zoals Wireshark.

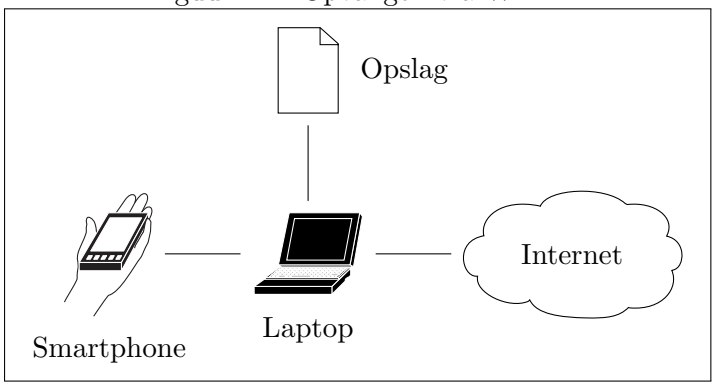
Voordeel van het deze methode is dat er geen opstelling gemaakt hoeft te worden om het netwerkverkeer af te vangen, dat gebeurt tenslotte in de telefoon zelf. Nadeel is dat het opgeslagen bestand nog verplaatst moet worden naar een pc om het te kunnen analyseren. Ook is het door de besturing en het formaat van een smartphone niet handig om hierop veel handelingen naast elkaar uit te voeren.

2.1.2 Wi-Fi

Vrijwel alle smartphones kunnen gebruik maken van draadloze netwerkverbinding door middel van Wi-Fi. Het is gemakkelijk om het netwerkverkeer af te vangen op het verbindingspunt van de Wi-Fi. De meeste andere onderzoeken naar smartphone-privacy gebruikten deze methode door een laptop als verbindingspunt te laten fungeren om vervolgens op de laptop het verkeer van de smartphone op te slaan.

Het voordeel van deze methode is dat dit werkt zonder enige aanpassingen aan de software op de smartphone. Een nadeel van deze methode is

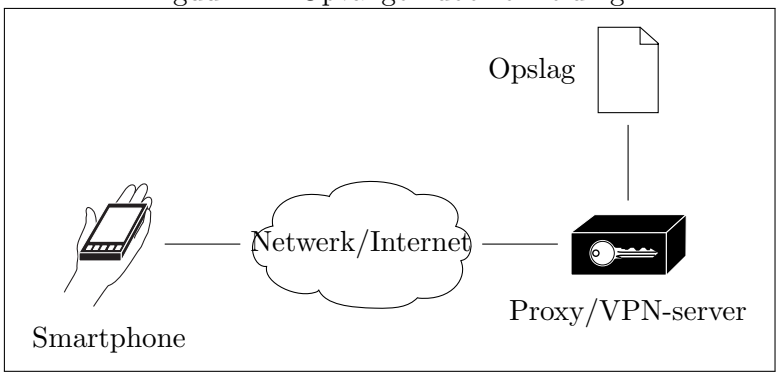
Figuur 2.1: Opvangen via Wi-Fi.



dat het alleen werkt in een lokaal netwerk en daarmee niet geschikt is om internetverkeer dat loopt via een telefoonmast (GSM/3G) af te vangen.

2.1.3 Netwerkverkeer omleiden

Figuur 2.2: Opvangen door omleiding.



Een andere mogelijkheid is om het netwerkverkeer om te leiden, bijvoorbeeld door gebruik van een proxy- of Virtual Private Network-verbinding (VPN). Bij deze methoden delegeer je het maken van verbindingen met het internet naar een andere computer. Deze computer laat je dan het netwerkverkeer dat afkomstig is van de smartphone opslaan om vervolgens te kunnen analyseren.

Proxy

Een proxy is een volmacht om namens een cliënt netwerkhandelingen uit te voeren. De cliënt stuurt zijn verzoek tot verbinding (bijvoorbeeld met een website) naar een proxyserver, die vervolgens de daadwerkelijke verbinding

legt. Na het leggen van een verbinding met een proxyserver gaat al het verkeer via die proxyserver.

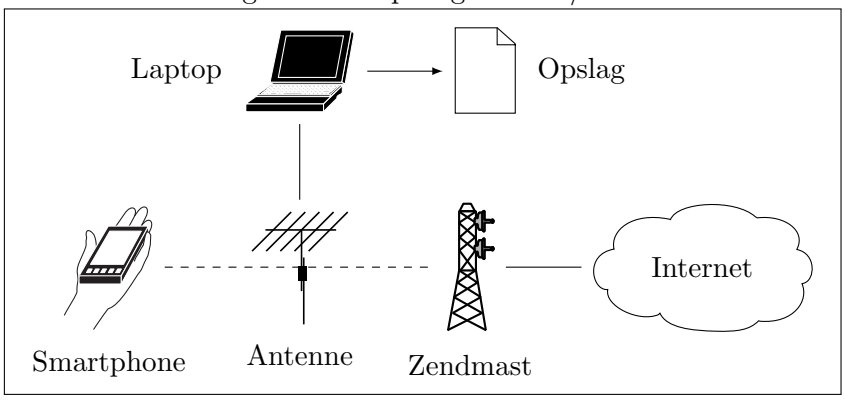
Nadeel van het gebruik van een proxyverbinding is dat niet alle besturingssystemen voor smartphones hier volledige ondersteuning voor hebben. Zo heeft Android alleen ondersteuning voor het gebruik van een proxy door de webbrowser, maar niet door andere apps. Om het probleem te omzeilen is root-toegang (volledige toegang tot het besturingssysteem) nodig wat aanvullende handelingen vereist.

Virtual Private Network

Een Virtual Private Network (VPN) is een privéverbinding tussen computers via een publiek netwerk. Een veel voorkomende situatie is dat werknemers vanuit thuis via een VPN verbinden met het netwerk van een bedrijf. Bijkomstigheid hiervan is dat uitgaande verbindingen met het internet vervolgens via de VPN-server lopen, waardoor het mogelijk wordt om op dit punt het netwerkverkeer op te slaan. Vrijwel alle besturingssystemen voor smartphones hebben een goede ondersteuning voor VPN-verbindingen.

2.1.4 GSM/3G afluisteren

Figuur 2.3: Opvangen GSM/3G.



Van den Broek (2011) onderzocht claims die meldden dat het mogelijk was om met relatief goedkope apparatuur GSM-verkeer af te luisteren. Hierin stelde hij dat het afluisteren praktisch onhaalbaar is, niet zozeer door de beveiliging van GSM, maar dat het opvangen van het verkeer lastig is en daarom dure apparatuur vereist. Bij 3G komt daarbij dat daar de beveiliging nog moeilijk te kraken is.

2.2 Protocollen

Bij het verkennen van de benodigdheden viel op dat bijna alle apps communiceren via HTTP-API's. Een API ofwel Application Programming Interface is een verzameling van voorgedefinieerde aanroepen waarmee een applicatie informatie kan verzenden en ontvangen. Met name HTTP in combinatie met XML of JSON wordt veel gebruikt door apps.

2.2.1 HyperText Transfer Protocol (HTTP)

HyperText Transfer Protocol wordt met name gebruikt voor de uitwisseling van websites tussen webservers en webbrowsers. In de context van dit onderzoek wordt het veel gebruikt door applicaties om gegevens uit te wisselen met API's. Het is een request/response-protocol waarbij de cliënt een request verstuurt naar een URI met daarbij een HTTP-header die extra informatie kan bevatten. Als de webserver het verzoek accepteert stuurt deze als respons een header terug, gevolgd door een body.

Figuur 2.4: Voorbeeld van HTTP-request.

```
GET /0.1/locations?lang=nl-NL&q=ams HTTP/1.1
User-Agent: Dalvik/1.4.0
(Linux; U; Android 2.3.7; HTC Desire Build/GRI40)
Host: api.9292.nl
Accept-Encoding: gzip
```

In figuur 2.4 wordt een verzoek gestuurd naar api.9292.nl met als inhoud /0.1/locations?lang=nl-NL&q=ams. In een browser ziet dit er uit als http://api.9292.nl/0.1/locations?lang=nl-NL&q=ams. Het is een verzoek aan de API van 9292ov.nl om suggesties te geven voor reisbestemmingen als er in het zoekveld in de app 'ams' is geschreven.

Figuur 2.5: Voorbeeld van een HTTP-response.

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/7.5
Date: Tue, 16 Apr 2013 12:37:25 GMT
Content-Length: 948

*Gevolgd door de body*
```

In het voorbeeld in figuur 2.5 geeft de webserver tekst in JSON-formaat terug als respons op het verzoek uit figuur 2.4. De respons bevat plaatsen waar iemand mogelijk heen zou willen reizen als hij 'ams' in het zoekveld ingevuld zou hebben. De eerste suggestie was Amsterdam-Centraal.

2.2.2 JavaScript Object Notation (JSON)

JavaScript Object Notation ofwel JSON is een lichtgewicht formaat om data uit te wisselen. Het is afgeleid van de methode om in JavaScript objecten te maken. Vanwege zijn compactheid wordt het veel gebruikt in API's.

Figuur 2.6: Voorbeeld van gegevens in JSON-formaat.

```
{
 "id": "station-amsterdam-centraal",
 "type": "station",
 "name": "Amsterdam Centraal",
 "place": {
  "name": "Amsterdam",
  "regionCode": "NH",
  "regionName": "Noord-Holland",
  "countryCode": "NL",
  "countryName": "Nederland",
 },
 "latLong": {
  "lat": 52.378706,
  "long": 4.900489
 },
 "urls": {
  "nl-NL": "/station-amsterdam-centraal",
  "en-GB": "/en/station-amsterdam-centraal"
 }
}
```

In figuur 2.6 werd 'station-amsterdam-centraal' als één van de locaties teruggegeven door de api van 9292ov.nl op het moment dat 'ams' in het tekstveld ingevoerd werd. Het bevat alle informatie die de app van 9292ov nodig heeft om de suggestie te weergeven.

2.2.3 eXtensible Markup Language (XML)

eXtensible Markup Language ofwel XML is net als JSON bedoeld voor de uitwisseling van gegevens. Het grote verschil met JSON is dat het XML meer mogelijkheden heeft om data te omschrijven. Nadeel van XML is dat de syntaxis meer ruimte inneemt. XML wordt net als JSON veelvuldig gebruikt door web-API's.

Figuur 2.7: Voorbeeld van gegevens in XML-formaat.

```
<location id="station-amsterdam-centraal" type="station">
 <place>
  <name>Amsterdam</name>
  <regionCode>NH</regionCode>
  <regionName>Noord-Holland</regionName>
  <countryCode>NL</countryCode>
  <countryName>Nederland</countryName>
 </place>
 <latlong>
  <lat>52.378706</lat>
  <long>4.900489</long>
 </latlong>
 <urls>
  <url lang="nl-NL">/station-amsterdam-centraal</url>
  <url lang="en-GB">/en/station-amsterdam-centraal</url>
 </urls>
</location>
```

Figuur 2.7 laat een mogelijke XML-implementatie zien van de suggesties van 9292ov in figuur 2.6.

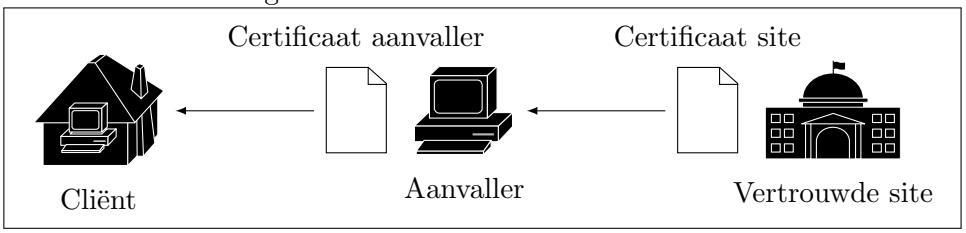
2.3 Ontsleutelen HTTPS

Tijdens het oriënteren naar een mogelijke proefopstelling viel mij op dat veel apps gebruik maken van een beveiligde verbinding door middel van Secure Sockets Layer (SSL). SSL is een versleutelingslaag tussen TCP/IP en een hoger gelegen protocol, in dit onderzoek vaak HTTP. Het heeft als doel de privacy en integriteit te waarborgen tussen twee communicerende applicaties. Een bekende toepassing hiervan zijn HTTPS-verbindingen, zoals tussen banken en rekeninghouders, met als doel dat je zeker weet dat je met de bank communiceert en transacties niet onderweg door een andere partij aangepast kunnen worden.

Een belangrijk deel van SSL/TLS is het gebruik van een public-key-infrastructuur met certificaten van Certificate Authorities. Het besturingssysteem of applicaties zelf houden een lijst bij met vertrouwde certificaten. Ieder certificaat dat een cliënt ontvangt (bijvoorbeeld van de bank) zal door een van deze vertrouwde certificaten ondertekend moeten zijn. Indien het certificaat niet ondertekend is krijgt de gebruiker een waarschuwing dat het onbetrouwbaar is en in sommige gevallen crasht een applicatie dan.

2.3.1 Man-in-the-middle-aanval

Figuur 2.8: Man-in-the-middle-aanval.



De meest voor de hand liggende methode om een HTTPS-verbinding te omzeilen is door gebruik te maken van een 'Man-in-the-middle'-aanval (MITM). Bij een MITM-aanval onderschep je het certificaat van de te vertrouwen website of API, en geef je de applicatie een zelfgemaakt certificaat. Met het eigengemaakte certificaat kun je het verkeer ontsleutelen, uitlezen en vervolgens weer versleutelen met het certificaat van de website of API waar het verkeer voor bestemd is. Deze aanval is mogelijk omdat je op zowel Android- als iPhone-besturingssystemen de vrijheid hebt om zelfgemaakte certificaten toe te voegen aan de lijst met vertrouwde certificaten. Daarbij is het belangrijk om te vermelden dat de aanval dus niet op afstand uitgevoerd kan worden.

Hoofdstuk 3

Proefopstelling

3.1 Meetcomputer

Om het netwerkverkeer van de smartphone op te vangen heb ik een aparte meetcomputer ingericht. Hierbij heb ik gebruik gemaakt van een virtuele machine (een softwarematige computer) door middel van VirtualBox met daarop Windows XP als besturingssysteem. Op die virtuele machine maakte ik gebruik van de applicaties Wireshark en CharlesProxy om het netwerkverkeer op te vangen.

De reden voor een virtuele machine is om een kaal besturingssysteem te hebben zonder interacties met mijn privé-applicaties. Door een aparte omgeving te creëren zorgde ik ervoor dat zo min mogelijk netwerkverkeer van een andere toepassing komt anders dan van de smartphone.

CharlesProxy en Wireshark zijn applicaties om het netwerkverkeer mee te monitoren. CharlesProxy is een zogenaamde HTTP-debugger. Hiermee kunnen makers van websites en webservices hun webserver testen op fouten. Omdat de meeste apps via het HTTP communiceren maken HTTP-debuggers het monitoren van het verkeer van smartphones zeer eenvoudig. CharlesProxy heeft zoals de naam doet vermoeden een ingebouwde proxy-server, waarmee de smartphone verbonden werd. Vervolgens ging al het verkeer van de smartphone via proxyserver van CharlesProxy waarna de inhoud van het verkeer bekeken kon worden. Ook kan het programma man-in-the-middle-aanvallen uitvoeren zodat SSL/HTTPS-verbindingen inzichtelijk worden. De keuze voor CharlesProxy is gebaseerd op persoonlijke voorkeur omdat ik het een gebruiksvriendelijke interface vind hebben. Alternatieven voor CharlesProxy zijn onder andere Fiddler, Burp Suite, Mitm-proxy en Mallory.

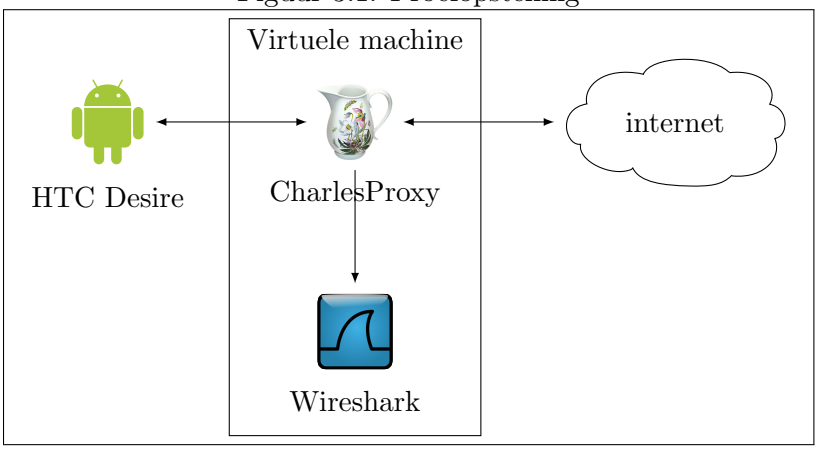
Wireshark is programma voor brede netwerkanalyse. Waar Charlesproxy geënt is op HTTP, is Wireshark algemener en geschikt voor een grote hoeveelheid protocollen. Wireshark gebruik ik als back-up voor het geval dat CharlesProxy niet goed overweg kan met het verkeer van een app.

3.2 Inrichten smartphone

De gebruikte smartphone is een HTC-Desire GSM met daarop een aangepaste versie van Android 2.3.7, namelijk Cyanogenmod 7.2.0. De reden voor het gebruik van een zogenaamde custom-rom is de root-toegang die daarbij komt. Root-toegang geeft je volledige rechten over het besturingssysteem van de telefoon. Dit was nodig voor het toevoegen van certificaten aan het besturingssysteem en het gebruik van ProxyDroid. ProxyDroid is een app om de smartphone met een proxyserver te verbinden zodat het netwerkverkeer van apps via de proxyserver gaat. Android heeft zelf ook wel de mogelijkheid om een proxy in te stellen, maar leidt vervolgens alleen het verkeer van de webbrowser via de proxyserver en niet het verkeer van de overige apps. ProxyDroid heeft ook de mogelijkheid om het netwerkverkeer van pps geïsoleerd naar de proxy te sturen. Hiermee kun je gemakkelijk het verkeer van de app afzonderen van het overige verkeer.

Het beveiligingscertificaat van CharlesProxy is toegevoegd aan het toestel door het bestand met de vertrouwde certificaten van de smartphone te downloaden naar een PC. Op de PC is het certificaat van CharlesProxy aan het bestand met vertrouwde certificaten toegevoegd en is het bestand vervolgens weer teruggeplaatst op de smartphone. IPhone en nieuwe versies van Android (vanaf 4.0) hebben standaard ondersteuning voor het installeren van certificaten, waardoor de tussenkomst van een pc niet nodig is.

Figuur 3.1: Proefopstelling



3.2.1 Normale gebruiker simuleren

Om te kunnen onderzoeken of apps persoonlijke gegevens versturen zal de smartphone persoonlijke gegevens moeten bevatten. Om de smartphone te laten gelijken op een reguliere smartphone heb ik de volgende gegevens toegevoegd:

- Contacten
- Foto's
- Google+ account
- Sim-kaart
- SMS-berichten
- Video's

Tijdens het testen werden stappen waarbij contacten van sociale media toegevoegd of benaderd zouden worden vermeden. Reden hiervoor is dat ik wilde voorkomen dat gegevens van echte personen in de testdata zouden komen. Om dit te omzeilen zouden er per sociaal medium verschillende dummy accounts aangemaakt moeten worden en deze met elkaar laten communiceren. Vanwege de tijd die het zou kosten om dummy accounts met elkaar te laten communiceren vallen accounts van sociale media buiten de scope van dit onderzoek.

3.3 Testprocedure

Om het netwerkverkeer van een app op te vangen volgde ik de volgende procedure:

1. App downloaden uit Google Play Store
2. Inhoud van het machtigingenscherm noteren
3. ProxyDroid instellen op app en verbinden met proxy
4. Per handeling in de app:
   (a) Nieuwe opname in CharlesProxy maken
   (b) Handeling uitvoeren
   (c) Opname opslaan
   (d) Beschrijving van handeling noteren
5. Verbinding met proxy verbreken
6. App verwijderen

In het geval dat het meten van het netwerkverkeer van een app niet goed ging met CharlesProxy verving ik CharlesProxy door Wireshark. De procedure bleef daarbij voor de rest gelijk.

Het noteren van de rechten die de app verzocht bij het installeren gaf de mogelijkheid om achteraf te kunnen controleren of apps niet meer verstuurden dan waar ze recht toe hadden. Door het noteren van de verschillende handelingen was het mogelijk om na te gaan of de verzonden data logischerwijs te verbinden viel met de handeling.

3.4 Analyse netwerkverkeer

Analyse van het opgeslagen netwerkverkeer gebeurde handmatig door het in CharlesProxy of Wireshark te bekijken. In het geval dat er in de HTTP-body nog coderingen werden aangetroffen werden deze, indien het type codering herkenbaar was, gedecodeerd. Als decoderen niet lukte werd er gekeken of het label iets vertelde over de inhoud en werd het label voor waarheid aangenomen. Indien een label ook geen uitsluitsel gaf over de inhoud werd de inhoud genoteerd als onbekend/versleuteld.

Hoofdstuk 4

Meetrapportage

4.1 Keuze van apps

Bij het meten van de apps is de proefopstelling gebruikt zoals beschreven in hoofdstuk 3. De keuze van de apps is deels gebaseerd op het onderzoek van The Wall Street Journal en op de masterscriptie van Emre Yildirim. Het overige deel is een verzameling van apps die ik zelf gebruik.

4.2 Overzicht meting

Uitkomsten van de metingen zijn te vinden in tabel 4.1. Daarbij een uitleg van de verschillende kolommen:

ID's zijn identificatienummers waarmee een gebruiker uniek geïdentificeerd kan worden. Binnen het Android- besturingssysteem bevinden zich verschillende soorten identificatienummers en ze verschillen met name in betrouwbaarheid en of er toestemming voor gevraagd dient te worden. Hierbij reken ik nummers zoals het identificatienummer van het besturingssysteem[1], MAC-adres van de WiFi-adapter en SIM-kaartnummers. Dit soort identificatienummers zijn interessant omdat hiermee unieke toestellen kunnen worden geïdentificeerd en gevolgd.

Gebruikersgegevens zijn gegevens die het identificeren van een persoon mogelijk maken. Denk hierbij aan namen, wachtwoorden en e-mailadressen. Wachtwoorden en e-mailadressen beschouw ik ook als gebruikersgegevens omdat deze vaak informatie over de gebruiker bevatten.

Onder **Locatie**gegevens beschouw ik gegevens waardoor de locatie van de telefoon vrij nauwkeurig te bepalen is. Hieronder vallen latitude/longitude-waarden, postcodes en plaatsnamen. Tijdzones en taal van het besturingssysteem beschouw ik er niet onder alhoewel deze ook een ruwe indicatie kunnen vormen van een locatie.

[1] http://developer.android.com/reference/android/provider/Settings.Secure.html#ANDROID_ID

Fingerprint is een verzameling van gegevens die losstaand een telefoon of gebruiker niet kunnen identificeren maar tezamen mogelijk een uniek profiel zouden kunnen creëren. Deze gegevens kunnen gebruikt worden om inzicht te krijgen op wat voor typen hardware en software de apps veelal gebruikt worden. De gegevens die hieronder vallen zijn hardware-specificaties zoals type toestel, schermresoluties en verbindingstype. Softwaregegevens die hieronder vallen zijn types en versies van het besturingssysteem, overzichten van geïnstalleerde apps en de naam van de app die gebruikt wordt. Ook valt hieronder informatie over de telecomprovider.

Encrypted data zijn de gevallen waarbij aanzienlijke hoeveelheden gegevens onleesbaar zijn door versleuteling of codering, of gevallen waarbij labels duidelijk aangeven dat het om versleutelde of gecodeerde gegevens gaat.

In tabel 4.1 zijn cellen blauw als dat type gegevens naar een domein werden verzonden dat van de maker van app is en zijn oranje gekleurd als dat type gegevens naar andere partijen verzonden werden. Het getal in de oranje cellen staat voor het aantal andere partijen waar de gegevens naar verzonden werden.

Tabel 4.1: Samenvatting van metingen

App	ID's	Gebruiker	Locatie	Fingerprint	Encrypted
9292					
Advanced Task Killer				1	
Alchemy				1	
Buienradar	3			3	
CBS News	1			5	
Daily Bible					
Dictionary.com	1		2	4	
Dumpert	1			2	
Facebook					
FOX News	2		1	4	
Fruit Ninja	4		1	4	
Google+					
Groupon	4			6	
Imdb	2		1	3	
inSSIDer				1	
Jewels	5	1		6	1

Voortzetting op volgende pagina

Tabel 4.1 – *Voortgezet van vorige pagina*

App	ID's	Gebruiker	Locatie	Fingerprint	Encrypted
Maps					
NLRadio					
NS Reisplanner Xtra	1			1	
NU.nl	3			4	
NYTimes for Android	1			18	
Paper Toss	2			2	
Shazam	4			3	
ShopSavvy Barcode Scanner	1			2	
The Weather Channel			2	3	
Toss It	1			2	1
TVGids.nl				1	
TwitchTV	1				
Wikipedia Mobile			1		
Yelp	1			2	
YouTube					

4.3 Analyse

Een patroon dat in tabel 4.1 naar voren komt is dat fingerprints van de smartphone vaak werden verzonden naar derde partijen. Blijkbaar is het identificeren van een telefoon/gebruiker interessant voor bepaalde partijen. Deze partijen lijken zich met name bezig te houden met het aanbieden van advertenties of met het inzichtelijk maken van hoe gebruikers navigeren door apps. Opvallend is dat de apps van bedrijven die zelf advertentieplatformen bieden, Facebook en Google, geen informatie naar derden sturen.

Verzending van gegevens zoals contactpersonen of foto's werden niet aangetroffen. Ook werden er geen gegevens aangetroffen waar geen toestemming voor was gevraagd in een machtigingenscherm.

4.3.1 Wegen van de meting

Tabel 4.1 geeft een objectieve weergave van welk type gegevens de verschillende apps verstuurden en waar naartoe. De tabel geeft niet weer of het opvragen en versturen van de gegevens een goede rede had. Tabel 4.2 is een weergave van de vraag of het versturen van een bepaald gegeven te koppelen valt aan de functie (vanuit de gebruiker geredeneerd) van de app. De kleur van een cel in de tabel is groen indien het verzenden van het gegeven nodig was voor de functionaliteit van de app en het niet werd gedeeld met partijen die hier niets aan bijdroegen. De kleur van een cel is rood als het versturen van het gegeven niet bijdroeg aan de functionaliteit, of als het gegeven gedeeld werd met een partij die niets toevoegde aan de functionaliteit. In geval van twijfel is een cel oranje.

Tabel 4.2: Weging noodzaak van verzending gegevens.

App	ID's	Gebruiker	Locatie	Fingerprint	Encrypted
9292					
Advanced Task Killer					
Alchemy					
Buienradar					
CBS News					
Daily Bible					
Dictionary.com					
Dumpert					
Facebook					
FOX News					
Fruit Ninja					
Google+					
Groupon					
Imdb					
inSSIDer					
Jewels					
Maps					
NLRadio					
NS Reisplanner Xtra					

Voortzetting op volgende pagina

Tabel 4.2 – *Voortgezet van vorige pagina*

App	ID's	Gebruiker	Locatie	Fingerprint	Encrypted
NU.nl					
NYTimes for Android					
Paper Toss					
Shazam					
ShopSavvy Barcode Scanner					
The Weather Channel					
Toss It					
TVGids.nl					
TwitchTV					
Wikipedia Mobile					
Yelp					
YouTube					

4.3.2 Apps uitgelicht

FOX News

Bij het aanklikken van het eerste nieuwsartikel dat getoond werd na het opstarten van de app van Fox News werden locatiegegevens verzonden naar een derde partij genaamd Nextap.[2] Als respons daarop kreeg ik een drietal suggesties voor nieuwsartikelen, maar geen van die artikelen leken enige relevantie te hebben tot mijn locatie.

Fruit Ninja

Na het spelen van een ronde in Fruit Ninja werden mijn locatiegegevens verzonden naar Beintoo. Beintoo is een bedrijf dat zich bezighoudt met klantbinding door middel van een virtuele munt genaamd 'bedollar'.[3] Met deze munt kan men aankopen doen of korting krijgen op producten van aangesloten retailers. Waarom bij deze stap locatiegegevens nodig zijn is niet op te maken uit het netwerkverkeer.

[2] http://www.nextap.co/
[3] http://www.beintoo.com/v2/how-it-works

Groupon

Groupon maakt gebruik van diensten van Crittercism[4] om de app sneller en beter te maken. Om dat te realiseren werden veel hardwaregegevens verzonden naar Crittercism waaronder batterijniveau, type processor en vrije ruime op intern en extern geheugen.

Google Maps

Google Maps valt op door afwezigheid van informatie. Ondanks dat de app de locatie van de smartphone leek te weten werden er geen locatiegegevens verzonden. Vermoedelijk bepaalt de app dan ook de locatie aan de hand van zichtbare Wi-Fi-routers in de buurt van de smartphone. In het verleden is Google in opspraak geweest wegens het verzamelen van Wi-Fi-gegevens met hun StreetView-auto's.[5]

NYTimes

Bij het lezen van een artikel in de app van de New York Times werd de mogelijkheid geboden om te abonneren. Na het klikken op de knop om te abonneren voerde de app een kettingreactie van verzoeken uit bij achttien verschillende bedrijven. Belangrijkste informatie die werd verzonden was het feit dat er op de knop was gedrukt om naar het abonnementenoverzicht te gaan.

Shazam

Shazam stuurde bij het opstarten het IMSI-nummer en het IMEI-nummer naar de eigen servers. Het IMSI is een nummer dat opgeslagen is in de SIM-kaart ter registratie van de SIM-kaart bij een mobiel netwerk.[6] Het IMEI-nummer is een identificatienummer voor telefoons op het mobiele netwerk. Net als de app van Groupon maakt Shazam ook gebruik van Crittercism en verstuurt allerlei hardware-informatie.

The Weather Channel

Bij verschillende acties in de app verstuurde The Weather Channel locatiegegevens naar het advertentiebedrijf Doubleclick[7] (onderdeel van Google) en naar Adobe[8]. Acties zoals het inzien van de weerkaart en verwachtingen

[4] http://www.crittercism.com

[5] http://tweakers.net/nieuws/74021/overheden-zetten-vraagtekens-bij-opslag-locatiedata-op-smartphone.html

[6] http://www.imsi.biz/

[7] http://www.google.com/doubleclick/

[8] http://www.2o7.net/

waren triggers hiervoor. Het ging hierbij om plaatsnaam en niet om exacte coördinaten.

Wikipedia Mobile

Wikipedia heeft de functie om artikelen over onderwerpen bij je in de buurt weer te geven. Hiervoor gebruikt het de database van GeoNames[9].

4.4 Validatie

4.4.1 Vergelijking WSJ

Van de apps uit het eerdere onderzoek van The Wall Street Journal[10] heb ik er zeventien getest. Van de zeventien apps heb ik bij vijf apps een soortgelijk resultaat. Bij vijf apps vond ik in mijn onderzoek meer gevallen van verzonden gegevens en bij zeven apps heeft het onderzoek van The Wall Street Journal er meer gevonden. Er zijn verschillende oorzaken voor het verschil denkbaar. Zo zou de testopstelling kunnen verschillen maar hier laat het artikel zich niet gedetailleerd over uit. Het is waarschijnlijk dat de onderzoeker een andere aanpak hanteerde tijdens het testen, door misschien meer of andere functies te gebruiken. In de uitleg van het artikel wordt gesteld dat de onderzoeker sociale media-functies heeft gebruikt, zoals vriendenzoekers. Sociale media-functies zijn overgeslagen in dit onderzoek.

4.4.2 Permission redelegation attack

Felt e.a. (2011) demonstreerde dat sommige apps vatbaar zijn voor redelegation-aanvallen. Hierbij fungeert een app die wel bepaalde machtigingen heeft als plaatsvervanger voor een app die de machtigingen niet heeft. In het geval van een aanval probeert een kwaadwillende app via een plaatsvervanger toch deze functies uit te voeren. Omdat ik alleen het netwerkverkeer gemeten heb van de app die ik op dat moment aan het testen was, is het overige verkeer van het besturingssysteem en andere apps niet meegenomen. Mocht er verkeer via dit type aanvallen zijn geweest, dan valt het onder verkeer van een andere app en werd dus niet meegenomen.

4.4.3 Overige zwakheden

De manier waarop deze apps zijn getest is een vorm van blackboxtesten. Bij blackboxtesten heeft de tester geen specificatie van de applicatie en kan hij alleen testen door te kijken wat de uitvoer is bij bepaalde invoer. De tester zal uiteindelijk nooit alle mogelijke invoer uitputtend kunnen testen,

[9] http://www.geonames.org/
[10] http://blogs.wsj.com/wtk-mobile/

omdat er simpelweg teveel invoermogelijkheden zijn en omdat de tester geen volledige invloed heeft op alle invoer. Apps zouden anders kunnen reageren op verschillende locaties, verschillende toestellen, verschillende sim-kaarten, verschillende tijdstippen en of er al eerder communicatie is geweest tussen app en server. Daarnaast worden apps continu geüpdatet, waardoor resultaten al snel verouderd zullen raken.

De gebruikte methode van testen valt en staat bij het kunnen lezen van informatie na het ontsleutelen van het HTTPS-verkeer door een man-in-the-middle-aanval. Ontwikkelaars van apps kunnen het analyseren van netwerkverkeer vertragen door binnen het HTTPS-verkeer nog een vorm van versleuteling toe te passen. Dit zal ervoor zorgen dat, om de versleuteling te verbreken, er overgegaan zal moeten worden tot reverse engineering (analyse van machine-/bytecode), een klus die meer tijd en vaardigheden vereist dan een man-in-the-middle-aanval.

Hoofdstuk 5

Conclusie

Versturen apps privacygevoelige gegevens en zo ja, waar sturen ze die dan heen? Om die vraag te beantwoorden heb ik in deze scriptie verschillende methoden bekeken om het netwerkverkeer van smartphones te onderscheppen. Vervolgens heb ik een proefopstelling gemaakt en deze toegepast op dertig verschillende apps. Hierbij keek ik naar de verzonden gegevens, de partijen waar deze gegevens naar toe werden gestuurd en of de verzonden gegevens nodig waren voor de functionaliteit van de app.

Uit het testen van de dertig apps bleek dat het versturen van privacygevoelige gegevens door apps eerder regel is dan uitzondering. Zo verstuurde bijna iedere app wel gegevens over de telefoon en verstuurde enkele apps ook gegevens over de gebruiker van de telefoon. In de meeste gevallen was er ook geen duidelijk verband tussen de verzonden gegevens en de functionaliteit van het programma. Het veelal niet aanwezig zijn van een functionele reden geeft dan ook sterk de indruk dat apps gebruikt worden als een medium voor het vergaren van gebruikersgegevens of het meten van het gedrag van gebruikers.

Dat apps privacygevoelige informatie versturen verbaasde mij niet. Wat mij wel verbaast is de mate waarin het gebeurt. Van de dertig apps waren er twee die helemaal niets verzonden. De overige apps, op drie na, verzonden in meer of mindere mate privacygevoelige informatie die niet nodig was voor het functioneren van de app.

Voor toekomstig onderzoek zou mijn meetopstelling gebruikt kunnen worden. Het is een goede methode om apps afzonderlijk te testen, maar de resultaten zullen afhankelijk zijn van hoe de onderzoeker de apps test met de opstelling. Voor het testen van grotere hoeveelheden apps zullen dan ook geautomatiseerde tests moeten komen. Er zou gedacht kunnen worden aan statische analyse van de byte- of machinecode van apps. Omdat het gedrag van de apps afhankelijk is van veel factoren hoop ik dan ook dat er veelvuldig onderzoek naar gedaan zal worden zodat er ten alle tijden een beeld is van het verzamelen van gegevens door middel van apps.

Bibliografie - Wetenschappelijk

[1] Fabian van den Broek. "Eavesdropping on GSM: state-of-affairs". In: *5th Benelux Workshop on Information and System Security (WISSec 2010)* (2011).

[2] Adrienne Porter Felt e.a. "Android permissions: user attention, comprehension, and behavior". In: *SOUPS*. Red. door Lorrie Faith Cranor. ACM, 2012, p. 3. ISBN: 978-1-4503-1532-6.

[3] Adrienne Porter Felt e.a. "Permission Re-Delegation: Attacks and Defenses". In: *USENIX Security Symposium*. USENIX Association, 2011.

Bibliografie - Overig

[4] Eric Smith. *iPhone applications & privacy issues: An analysis of application transmission of iPhone unique device identifiers (UDIDs)*. Tech. rap. Technical Report, 2010.

[5] Emre Yildirim. "Mobile Privacy: Is There An App For That?" Masterscriptie. University of Amsterdam, 2012.