LUNDS UNIVERSITET

Ekonomihögskolan

Säkerhet vid utveckling av mobilapplikationer, bortglömd eller prioriterad?

Kandidatuppsats, 15 högskolepoäng, SYSK02 i Informatik

Framlagd: 2013-06-17
Författare: Mirza Begic
Paul Radutiu
Handledare: Anders Svensson
Examinatorer: Björn Johansson
Nicklas Holmberg

Titel: Säkerhet vid utveckling av mobilapplikationer, bortglömd eller prioriterad?

Författare: Mirza Begic
Paul Radutiu

Utgivare: Lunds Universitet, Institutionen för Informatik

Handledare: Anders Svensson

Examinatorer: Björn Johansson
Nicklas Holmberg

Publiceringsår: 2013

Uppsatstyp: Kandidatuppsats

Språk: Svenska

Nyckelord: Mobil säkerhet, Mobilapplikation, Informationssäkerhet, Risker, Utveckling

Abstrakt

Det finns inte mycket studier kring säkerhet vid utveckling av mobilapplikationer trots att säkerheten i mobilapplikationer håller på att få en viktigare roll i samhället. Det finns en mängd säkerhetsrisker vad gäller mobilapplikationer. Dessa risker kan leda till stor skada om de drabbar känslig data i applikationen. Denna studie undersöker hur säkerhet införs vid utveckling av mobilapplikationer hos IT-konsultföretag i Malmö och Lund och hur arbetet med säkerhet är utformat. Vi har använt oss av böcker och vetenskapliga artiklar för att identifiera de viktigaste faktorerna som bör finnas med i utveckling av säkerhet i mobilapplikationer och skapat ett teoretiskt ramverk som är grunden till vår studie. Den empiriska undersökningen i denna studie består av intervjuer med fyra företag. I analysen i studien jämfördes den insamlade datan med vår litteratur samt strukturerades upp efter vårt teoretiska ramverk. Det har visat sig att konsultföretagen inkluderar majoriteten av de viktiga faktorer som tas upp i litteraturen. Det finns dock skillnader i hur dessa faktorer inkluderas och i vilken omfattning de inkluderas. Vår studie visar att företagen har kompetens till att skapa säkra applikationer men att deras arbete med säkerhet endast begränsas av kunders ovillighet att inkludera säkerhet i mobilapplikationer. Följaktligen så är mobilapplikationer säkra endast om de som betalar för utvecklingen vill att de ska vara det.

Förord

Vi skulle vilja tacka Anders Svensson, vår handledare, för det stöd och konstruktiv kritik som han har bidragit med. Vi skulle även vilja tacka våra respondenter för deras tid och hjälp, utan de skulle uppsatsen inte vara möjlig. Ett särskilt tack vill vi ge till Himzo Music för hjälp med kontaktuppgifter. Slutligen vill vi tacka alla som på något sätt har bidragit med kunskap och råd, däribland Lars Fernebro.

Innehållsförteckning

1 Inledning

1.1 Bakgrund

Användningen av smarta mobiltelefoner (smartphones) har ökat exponentiellt under de senaste åren. Enligt amerikanska IT-forsknings- och rådgivnings företaget Gartner (2012) så såldes det under 2012, 207,7 miljoner smarta mobiltelefoner. Eftersom möjligheterna med dagens smarta mobiltelefoner är många så förlitar sig människor alltmer på sina smarta mobiltelefoner när det gäller att utföra handlingar som att hålla sig uppdaterade med den senaste informationen, sköta betalningar och överföringar, spela spel, uppdatera sin status på de olika sociala medierna som finns etc. Dagens smarta telefoner är helt enkelt skärningspunkten mellan persondator och mobiltelefon (Ramu, 2012). Då dessa enheter fick sin genomslagskraft för ett antal år sen har också ett nytt fenomen nästan exploderat, nämligen mobilapplikationer som i folkmun kallas för "appar". Nya mobilapplikationer släpps dagligen och görs tillgängliga för nedladdning till konsumenter, och det finns idag mobilapplikationer för i princip alla möjliga ändamål, ändamål som många gånger kan vara kopplade till olika typer av känslig information som exempelvis bankinformation och personuppgifter (Basavala, Kumar & Agarrwal, 2013).

1.2 Problemformulering

Vid en första anblick så har användandet av smarta mobiltelefoner och dess många olika mobilapplikationer gett många positiva möjligheter för användare, främst när det gäller mobiliteten och alla funktioner som är möjliga att göra. Dock så finns det en baksida av detta mynt och faktum är att antalet virus och andra skadliga programvaror riktade mot dessa enheter växer lavinartat, och om fel information kommer i fel händer kan det få dystra konsekvenser (Sujihtra & Padmavathi, 2012). Under det första kvartalet för 2013 publicerade det finska antivirusföretaget F-secure (2013) sin mobila hotrapport där det framkom att man hade identifierat sammanlagt 149 familjer + varianter av hot, en ökning med 49% från förra kvartalet. Trots detta så har tyvärr utvecklingen av mobilapplikationer drivits mer av den stora efterfrågan på marknaden, med fokus på nya funktioner, medan säkerheten försummats (Elfattah, Youssif & Ahmed, 2011). Företagen som utvecklar dessa applikationer har därför ett stort ansvar att leverera säkra mobilapplikationer till konsumenterna. Om dessa företag inte tar hänsyn till säkerhet så finns risken att slutprodukterna inte är säkra och att användarnas mest känsliga information hamnar i fel händer (Son, Lee & Oh, 2012; Ramu, 2012).

1.3 Forskningsfråga

Utifrån det som hittills har nämts har vi formulerat vår forskningsfråga.

Vår forskningsfråga är:

"På vilket sätt inför företag säkerhet vid utvecklingen av mobilapplikationer?"

1.4 Syfte

Denna uppsats belyser på vilket sätt IT-konsultföretag inför säkerhet vid utveckling av mobilapplikationer. Uppsatsen undersöker de åtgärder som dessa företag tar för att införa säkerhet i de mobilapplikationer som de utvecklar. Syftet med detta är att, genom att veta hur de arbetar med säkerhet, ta reda på hur högt säkerheten prioriteras under utvecklingen av mobilapplikationer.

1.5 Avgränsningar

För att inte göra vår studie för omfattande, och på så sätt ohanterlig, så har vi valt att hålla oss inom vissa gränser. Först och främst så kommer vi endast att behandla mobilapplikationer som installeras på mobilenheter och existerar i en specifik plattform vilket innebär att vi inte kommer att skriva om, exempelvis, webbapplikationer. Vi har även valt att, i vår empiriska undersökning, hålla oss till IT-konsultföretag som utvecklar mobilapplikationer i Malmö och Lund. När det gäller säkerhet så kommer vi endast att behandla säkerhet i den kontext där mobilapplikationer utsätts för attacker direkt från angripare, vilket innebär att vi inte kommer att ta upp de risker då angriparen manipulerar användaren av en mobilapplikation för att utföra en attack. Någonting som tydligt framgår, men dock måste nämnas, är att vi inte kommer att ta upp hur utvecklare bör utveckla säkerhet på annat håll utöver i mobilapplikationer.

2 Litteraturgenomgång

För att svara på vår forskningsfråga så måste vi först och främst anskaffa oss en teoretisk grund. Vår teoretiska grund består utav åtta vetenskapliga studier inom mobil säkerhet samt tio välkända böcker inom mobil säkerhet som är skrivna av olika kunniga personer med skilda bakgrunder. Vi har valt dessa källor som grund för vår teori eftersom de alla hanterar mobil säkerhet, fast ur olika perspektiv, vilket ger oss en omfattande bild av mobil säkerhet. Denna teori kommer att analyseras och de mest frekvent uppkommande faktorerna gällande hur man utvecklar säkra mobilapplikationer kommer att användas av oss i utformningen av vårt teoretiska ramverk. Vi kommer även att presentera de olika faktorerna utifrån vad vår litteratur säger om dessa. Utifrån detta ramverk kommer vi senare att forma vår empiriska undersökning samt vår analys. I de fall där flera källor tar upp samma påstående har vi valt att endast presentera en av dessa källor i litteraturgenomgången, för att undvika redundans.

2.1 Utformning av teoretiskt ramverk

Vi har analyserat vår litteratur och summerat frekvensen av de faktorer som, i vår litteratur, anses vara viktigast för utvecklare när de ämnar skapa säkra mobilapplikationer. Dessa nio faktorer utgör tillsammans det teoretiska ramverk som vi kommer att använda oss av i vår studie. Detta ramverk kommer sedan att forma vår empiriska undersökning och vår analys av denna undersökning.

Vi har valt att dela in de faktorer som vi har hittat i tre kategorier. Dessa tre kategorier nämns inte i litteraturen. Vi har själv skapat dessa kategorier för att göra det enklare att förstå och överblicka de olika faktorer som vi har funnit i vår litteratur samt för att ge vårt teoretiska ramverk en grundläggande struktur.

Viktiga faktorer gällande tekniska åtgärder för att skapa säkra mobilapplikationer						
Referens		**Kryptering av kommunikation**	**Autentisering**	**Säker lagring av data**	**Validera input**	**Testa säkerheten**
Akademiska artiklar	Basavala, Kumar Agarrwal (2013)	X	X	X	X	X
	Becher et al. (2011)	X	X	X	X	-
	La Polla, Martinelli & Sgandurrara (2012)	X	X	X	-	X
	Ramu (2012)	X	X	X	-	-
	Rassan & Al Sheikh (2013)	X	X	X	-	-
	Son, Lee & Oh (2012)	-	-	-	X	X
	Sujithra & Padmavathi (2012)	X	X	X	X	-
	Wang, Streff & Raman (2012)	X	X	X	-	-
Böcker	Dunham (2008)	X	X	X	X	X
	Dwivedi (2010)	X	X	X	X	X
	Fried (2010)	X	X	X	X	X
	Gunasekera (2012)	X	X	X	-	X
	Hoog (2011)	X	X	X	-	X
	Hoog & Strzempka (2011)	X	X	X	-	X
	Kadrich (2007)	X	X	X	X	X
	Miller et al. (2012)	X	X	X	X	X
	Six (2011)	X	X	X	X	X
	Zdziarski (2012)	X	X	X	X	X
Summa:	18	18	17	17	11	10

Tabell 1: Frekvensen av tekniska åtgärder som nämns i litteraturen.

Viktiga faktorer gällande säkra arbetssätt för att skapa säkra mobilapplikationer			
Referens		**Säkra praxiser**	**Användning av verktyg**
Akademiska artiklar	Basavala, Kumar Agarrwal (2013)	-	-
	Becher et al. (2011)	-	-
	La Polla, Martinelli & Sgandurrara (2011)	X	-
	Ramu (2012)	X	-
	Rassan & Al Sheikh (2013)	-	-
	Son, Lee & Oh (2012)	X	X
	Sujithra & Padmavathi (2012)	-	-
	Wang, Streff & Raman (2012)	-	-
Böcker	Dunham (2008)	X	-
	Dwivedi (2010)	X	X
	Fried (2010)	X	-
	Gunasekera (2012)	X	X
	Hoog (2011)	X	X
	Hoog & Strzempka (2011)	X	X
	Kadrich (2007)	X	X
	Miller et al. (2012)	X	X
	Six (2012)	X	X
	Zdziarski (2012)	X	X
Summa:	18	13	9

Tabell 2: Frekvensen av säkra arbetssätt som nämns i litteraturen.

Viktiga faktorer gällande säker kultur för att skapa säkra mobilapplikationer			
Referens		**Säkerhetsmål**	**Kunskap**
Akademiska artiklar	Basavala, Kumar Agarrwal (2013)	X	-
	Becher et al. (2011)	-	-
	La Polla, Martinelli & Sgandurrara (2011)	X	-
	Ramu (2012)	X	-
	Rassan & Al Sheikh (2013)	-	X
	Son, Lee & Oh (2012)	-	X
	Sujithra & Padmavathi (2012)	-	-
	Wang, Streff & Raman (2012)	-	-
Böcker	Dunham (2008)	X	-
	Dwivedi (2010)	X	-
	Fried (2010)	X	X
	Gunasekera (2012)	X	-
	Hoog (2011)	X	-
	Hoog & Strzempka (2011)	X	-
	Kadrich (2007)	X	X
	Miller et al. (2012)	X	-
	Six (2012)	X	-
	Zdziarski (2012)	X	-
Summa:	18	13	4

Tabell 3: Frekvensen av säkra kulturella faktorer som nämns i litteraturen.

2.2 Viktiga tekniska åtgärder

Vi har valt att placera alla frekvent omnämnda faktorer som hanterar någon form av specifik teknisk åtgärd i denna kategori. Dessa faktorer hanterar endast en typ av teknisk åtgärd och inte en samling av flera typer av tekniska åtgärder.

Som det tydligt går att se i tabell 1 så är alla tekniska åtgärder, som utvecklare bör använda för att skapa säkra applikationer, omnämnda i vår litteratur. Nedan förklaras de olika åtgärderna utifrån vad vår litteratur säger om dem.

2.2.1 Kryptering av kommunikation

Rörlig data är detsamma som rörlig risk. Det är enklare att skydda någonting som inte rör sig än någonting som transporteras från en punkt till en annan (Fried, 2010). Känslig data som skickas till och från mobilapplikationer måste vara svårläst för en angripare. Ett sätt att göra så att angripare inte kan avläsa dessa data är genom kryptering (Gunasekera, 2012). Data som inte är krypterad kan fångas upp enkelt (Hoog, 2011) och enligt La Polla, Martinelli & Sgandurra (2012) så kan grundläggande kryptering öka kvaliteten på en applikations säkerhet till stor del. Wang, Streff & Raman (2012) skriver att kryptering är enkelt att använda sig av. En mobilapplikation som kommunicerar gör det ofta med en server. Känslig data kan ligga i eller skickas till servern, vilket gör att data som transporteras till och från servern kan avlyssnas eller manipuleras om den inte skyddas (Kadrich, 2007). Även data som inte skickas till en server, utan till andra enheter kan avlyssnas och manipuleras av angripare (Rassan & Al Sheikh, 2013). På grund av de konsekvenser som kan uppstå ifall data inte krypteras medan den skickas så bör all känslig data som skickas till och från applikationen krypteras (Ramu, 2012). Sujithra & Padmavathi (2012) nämner att utvecklare i vissa fall väljer att återgå till okrypterad kommunikation när angripare bryter sig igenom krypteringen. De menar att detta måste undvikas.

Kadrich (2007) skriver om klientsäkerhet och hur viktigt det är att skydda klienterna av en server. Han menar att mobilapplikationer bör skyddas, men en av de mer viktiga punkterna när det gäller klientsäkerhet är att se till så att servern är skyddad från applikationen. Eftersom servern oftast innehåller känslig data och eftersom applikationen oftast har direkt tillgång till servern så är det viktigt att se till att servern är skyddad från applikationen. Detta kan göras genom att ha säkerhetsmekanismer i servern, men man kan även, och bör, använda sig av vissa åtgärder i applikationen så som kryptering. Dunham (2008) hänvisar till detta som end-to-end encryption eller traffic encryption. Dunham (2008) nämner även att, ur serverns perspektiv, så blir datan mer pålitlig om den skickas krypterad. Miller et al. (2012) skriver att ur applikationens perspektiv så blir datan också mer pålitlig om den skickas krypterad.

Dock måste krypteringen vara stark för att förhindra angripare från att dekryptera datan med hjälp av brute force attacker (Six, 2011). Att använda sig av dålig kryptering kan leda till att applikationen hanterar den data som tas emot, som om den vore säker. Denna datan hade kanske hanterats på ett annat sätt, på grund av säkerhetskäl, ifall den inte skickades krypterad. På grund av att den var krypterad men på ett dåligt sätt så hanterar applikationen datan som om den vore säker, trots att den inte är så säker som applikationen antar (Miller et al. 2012). Kadrich (2007) nämner även att det inte bara är viktigt att kryptera data som skickas, utan även data som lagras. Han skriver att känslig data bör lagras krypterad på den mobila enheten. Han påpekar att det finns diverse verktyg på marknaden som kan underlätta för utvecklare när det gäller kryptering. Dock måste utvecklare ta hänsyn till vad det är för applikation som

utvecklas. En applikation som inte kommunicerar med en server eller som inte hanterar känslig data behöver inte ha dessa säkerhetsmekanismer inbyggda (Kadrich, 2007). Zdziarski (2012) instämmer med de andra författarna när han skriver att kryptering är bland de bästa sätten för en applikation att skydda känslig data och det viktigaste, enligt honom, är att kryptering implementeras på rätt sätt. Hoog & Strzempka (2011) påpekar att många utvecklare inte tar hänsyn till säkerhet och krypterar inte på rätt sätt. De menar att utvecklare inte bör vara mindre intresserade av säkerhet än angripare.

Six (2011) understryker vikten av att distribuera säkerhetsnycklar på ett säkert sätt så att angripare inte har tillgång till dessa, vilket i värsta fall kan främja en fruktad och allvarlig attack mot kommunikationen, en mannen-i-mitten attack. En teknik för säker kommunikation som ofta förekommer i vår litteratur är SSL. Dunham (2008) skriver att SSL är den vanligaste tekniken som utvecklare använder sig av för att skapa säker kommunikation och att den också är enkel att använda sig av. Kadrich (2007) nämner, utöver SSL, även HTTPS och TLS. Även Basavala, Kumar & Agarrwal (2013) nämner HTTPS som en vanlig standard vid säker kommunikation. Dessa standarder finns i de flesta mobilenheter och de har blivit de mest använda i branschen. Dock så brukar den inbyggda säkerheten i mobilenheterna vara det första angripare attackerar och på grund av detta så bör utvecklare, om datan är känslig, inte enbart förlita sig på den kryptering som finns inbyggd i mobilenheterna (Dwivedi, 2010). Gunasekera (2012) skriver att SSL är säkert men att den inte är säker nog. Dwivedi (2010) menar att utvecklare bör göra ytterligare insatser för att skapa säker kommunikation ifall att de befintliga standarderna skulle fallera under en attack.

Becher et al. (2011) skriver att även om kryptering av kommunikation är av största vikt så påverkas samtidigt batteritiden i enheten av kryptering. Ju mer man krypterar desto mer energi krävs det av den mobila enheten, vilket kan resultera i mindre batteritid. Detta är någonting som utvecklare måste ta hänsyn till och vara beredda på.

2.2.2 Autentisering

Det finns en till faktor när det gäller säker kommunikation. Det är viktigt för en applikation att säkerställa att den som tar emot datan som skickas från applikationen eller den som skickar datan som tas emot av applikationen verkligen är den som den utger sig för att vara. Detta görs med hjälp av autentisering (Dwivedi, 2010). Det är viktigt för mobilapplikationer att, under kommunikation, använda sig av autentisering (Fried, 2010). Om en applikation kommunicerar känslig data så är det viktigt för den att säkerställa att mottagaren verkligen har tillstånd till att ta del av dessa känsliga data. Detta är ett försök att förhindra angripare från att komma åt känslig data. Det är även viktigt för applikationer att säkerställa att den som skickar känslig information till applikationen har tillåtelse att göra detta, eftersom felaktig data kan skickas och på så sätt kan en attack mot applikationen framkallas (Dwivedi, 2010). Wang, Streff & Raman (2012) påpekar att autentinsering är en av de tre säkerhetsegenskaper som är mest önskade idag. Basavala, Kumar & Agarrwal (2013) skriver att bra autentisering inte används så går det inte att skydda känslig data. Rassan & Al Sheikh (2013) påpekar att det är viktigt att använda sig av autentisering både när det gäller användare av applikationen och annan mjukvara som kommunicerar med applikationen.

Det finns många sätt att autentisera på och ett av de mer enkla sätten är genom lösenord. Det är ett sätt för användaren av applikationen att bevisa att denne får lov att ta del av den känsliga datan som applikationen har tillgång till. Ett mer avancerat sätt att autentisera sig på är genom att signera sin kod. En signerad kod visar att den som har skrivit koden är pålitlig

och det visar även att koden inte har manipulerats av en opålitlig källa (Dunham, 2008). Kadrich (2007) instämmer med Dunham (2008) och fortsätter med att förklara att det är viktigt för applikationer att fråga om lösenord. En angripare kan dölja sin identitet och maskera sig som en pålitlig användare. Kadrich (2007) skriver att angripare förhindras från att göra detta om applikationen frågar efter lösenord.

Dock måste lösenordet vara säkert. Ett säkert lösenord är, enligt författaren, ett lösenord som är svårt att gissa sig fram till med hjälp av brute force attacker. Zdziarski (2012) styrker detta genom att förklara att lösenord bör innehålla stora och små bokstäver samt siffror för att säkerställa att de är så säkra som möjligt. Eftersom Kadrich (2007) skriver om hur servern bör skyddas från applikationen så påpekar han att applikationen måste autentisera sig varje gång den ska ha tillgång till servern. Six (2011) instämmer genom att skriva att det första steget i kommunikationen mellan en applikation och en server bör vara autentisering. Six (2011) menar att, ur applikationens perspektiv, så är det viktigt att veta att det är rätt server applikationen kommunicerar med eftersom en angripare kan påstå att de är en server och på så sätt skicka skadlig data till applikationen. Hoog (2011) fortsätter på samma spår och skriver att det även är viktigt att verifiera de certifikat som används på ett rätt sätt. Han skriver att man inte bör lita på någon för att de har ett certifikat utan man måste, på ett bra sätt, kontrollera att det är rätt certifikat. Hoog & Strzempka (2011) understryker vikten av att verifiera certifikat för att skydda sig mot mannen-i-mitten attacker.

De certifikat som används för autentisering måste vara pålitliga, enligt Kadrich (2007), vilket innebär att utvecklare bör använda sig av kända och pålitliga tekniker för autentisering. Miller et al. (2011) skriver att de certifikat som utvecklare använder sig av bör komma från pålitliga certifikatutfärdare. Dessa kan även, enligt Gunasekera (2012) vara egna certifikat. SSL, som tidigare nämnts, innehåller även autentisering och denna teknik brukar vara pålitlig (Six, 2011). Zdziarski (2012) beskriver SSL som säkert men menar att sekundära åtgärder för säker kommunikation säkerställer att data skickas på ett säkert sätt. Om ett SSL certifikat skulle vara osäkert under en attack så kan sekundära åtgärder, i form av egna certifikat, säkerställa att datan som skickas inte manipuleras eller blir avläst av angripare. Miller et al. (2012) påpekar att en tvåfaktorsautentisering är en bra egen mekanism att använda sig av. Dock skriver Ramu (2012) att en Zitmo attack has lyckats bryta sig igenom bankers tvåfaktorsautentisering. Sujithra & Padmavathi (2012) nämner att utvecklare både borde använda sig av teckenbaserad autentisering och kunskapsbaserad autentisering.

Becher et al. (2011) påpekar att det även är viktigt att spara autentiseringsinformation på rätt sätt. Om en angripare kommer åt autentiseringsinformationen så spelar det ingen roll om certifikaten är pålitliga eftersom angriparen, med denna information, har rätt att komma åt känslig data. La Polla, Martinelli & Sgandurra (2012) ger ett exempel på detta när de skriver om angripare som har kommit åt privatpersoners bankkonton genom SMS eftersom de hade tillgång till autentiseringsinformationen.

2.2.3 Säker lagring av data

Kommunikation, eller transportering av data, öppnar upp för hot från angripare och den mest fruktade är mannen-i-mitten attacker (Zdziarski, 2012). Men data är även sårbar när den inte transporteras, utan när den lagras (Dwivedi, 2010). Mobil malware har ofta som mål att komma åt känslig data, även när den lagras (La Polla, Martinelli & Sgandurra, 2012). Mobilapplikationer lagrar mer och mer data lokalt istället för i servern vilket innebär att det blir viktigare att lagra data säkert i applikationen. Rassan & Al Sheikh (2013) instämmer med

detta och skriver att en anledning till att säker lagring av data är viktigt är för att mer och mer känslig data hanteras och sparas i applikationer, så som exempelvis information om bankkonton. En av de viktigaste faktorerna som påverkar säker lagring av data är beslutet av var data ska lagras. Utvecklare måste tänka noggrant på var de lagrar datan. Ett exempel är att de bör undvika att lagra data i externa minnen så som minneskort, eftersom alla mobilenheter inte tillåter kryptering av data i dessa (Dwivedi, 2010). Wang, Streff & Raman (2012) instämmer med detta och föreslår att känslig data sparas i molnet istället för lokalt. Ett annat exempel som Fried (2010) tar upp är att data kan lagras på två platser. Han menar att hälften av datan kan lagras på en plats och andra hälften på en annan plats. Det gör att angriparen inte kommer åt all data ifall viss data blir attackerad. Hoog & Strzempka (2011) föreslår att man, i så stor grad som det är möjligt, undviker att lagra känslig data. Med detta menar de att data som inte behöver lagras inte ska lagras, om den är känslig. Deras motivering till detta förslag är att angripare inte kan komma åt data som inte existerar. De exemplifierar detta genom att säga att kontonummer inte behöver lagras i applikationen bara för att man kommunicerar med en bank. Becher et al. (2011) nämner även ett exempel där vissa applikationsutvecklare väljer att använda sig av dataspegling, där all data i applikationen kopieras till ett externt minne utanför den mobila enheten. De rekommenderar att detta inte görs ifall det inte går att vara säker på att det externa minnet har god säkerhet. Detta är på grund av att angripare kan attackera applikationens data utan att attackera applikationen.

Ramu (2012) skriver att all data som lagras lokalt och inte på servern måste krypteras och Zdziarski (2012) instämmer och fortsätter med att skriva att kryptering måste, som tidigare nämnts, vara starkt. Basavala, Kumar & Agarrwal (2013) skriver att skydda data innebär att kryptera data. Hoog & Strzempka (2011) påpekar att enheters egna lagringssystem i många fall kan räcka till, men det är viktigt att det används på rätt sätt. Man kan få en falsk känsla av säkerhet om man förlitar sig på enhetens egna säkerhet, men inte använder den på rätt sätt. Miller et al. (2012) anser att enheters egna lagringssystem ofta inte räcker till. Zdziarski (2012) instämmer genom att påpeka att mobilenheternas egna lagringsfunktioner är säkra, men på grund av att angripare oftast attackerar enhetens inbyggda säkerhet först så går det inte att garantera att datan är säker genom att enbart förlita sig på enhetens egna lagring. Han menar att utvecklare, om datan verkligen är känslig, bör utföra egen kryptering av datan.

Utöver att all data som lagras i enheten bör krypteras och utöver att det är viktigt att tänka på var känslig data lagras så skriver Six (2011) att det även är viktigt att tänka på om data behöver hämtas eller inte. Han understryker betydelsen av att utvecklare bör undvika att hämta och skicka känslig data så mycket som möjligt eftersom sannolikheten att angripare kommer över känslig data ökar i samband med ett ökat antal platser där dessa känsliga data existerar. Han exemplifierar detta genom att skriva att känslig data inte behöver hämtas om detta endast görs för att bevisa att en annan entitet, utöver applikationen, har tillgång till samma data som applikationen. Detta kan bevisas på andra sätt, exempelvis genom autentisering. Om känslig data tvunget måste hämtas för att jämföra med en annan entitets känsliga data så går det att göra detta genom att en hashversion av den känsliga datan jämförs med den andra entitetens hashversion av datan, viket är säkrare än att jämföra originalversioner av datan (Six, 2011).

Kadrich (2007) skriver att det är viktigt att skydda data i applikationen av två anledningar. Den första som han nämner är att data som finns i en applikation kan komma från en server. Oavsett hur skyddad en server är så är datan inte skyddad om den lagras på ett sätt som inte är säkert av applikationen. Det är därför viktigt att se till så att datan som lagras av applikationen

i mobilenheten är skyddad, eftersom den inte längre är i serverns kontroll. Den andra anledningen som Kadrich (2007) nämner är att känslig data kan genereras av en applikation och denna datan brukar sparas i själva applikationen. Även om denna datan inte existerar i servern så räknas det som känslig data och måste därför skyddas genom att lagras på ett säkert sätt. Han skriver även att man, som utvecklare, bör fokusera på att ha så mycket som möjligt av den känsliga datan i servern och inte mobilapplikationen.

Trots att säkerhet oftast gör att det tar längre tid för människor att komma åt deras egna data så är det ett pris som måste betalas, eftersom angripare annars kan skapa mer besvär än den extra tid som det tar att hämta en bit data (Hoog, 2011). Gunasekera (2012) anser att säkerhet, även om det är svårt att skapa, är väldigt viktigt.

2.2.4 Validera input

Även om datan lagras krypterad och på en säker plats så innebär det inte att datan är säker från angripare (Dwivedi, 2010). Son, Lee & Oh (2012) nämner att attacker som skett med hjälp av skadlig input tidigare har orsakat stora ekonomiska skador. Basavala, Kumar & Agarrwal (2013) skriver att alla användare av en applikation inte är pålitliga. Angripare kan enkelt utföra en attack från applikationen genom att skicka skadlig input genom applikationen till servern. Två exempel på hur angripare kan komma åt krypterad data är SQL-injection och buffer overflow. Alla dessa attacker kan ske på grund av att utvecklare inte kontrollerar den input som applikationen tar emot (Dwivedi, 2010). Även Becher et al. (2011) skriver att angripare kan få tillgång till känslig data om utvecklare utelämnar validering av input. Dunham (2008) skriver om hur angripare kan attackera applikationer genom input på många olika sätt och beskriver hur detta möjliggörs av slarvig kod som inte validerar den input som applikationen tar emot. Även Kadrich (2007) visar hur farligt det kan vara ifall input inte valideras. Speciellt för servern som är mottaglig för SQL-injections. Kadrich (2007) beskriver även hur angripare kan få tillgång till data utanför den databas som applikationen associeras med och hur detta enkelt kan motverkas med hjälp av att validera input. Även Fried (2010) påpekar att säkerheten i mobilapplikationer, till stor del, förlitar sig på att input valideras. Han menar att validering bör och förväntas finnas i mobilapplikationer.

För att förhindra dessa typer av attacker, som förlitar sig på att utvecklaren inte validerar input så föreslår Six (2011) att utvecklare måste skapa applikationer som antar att all input är farlig. Utvecklare bör alltid anta att all input har manipulerats av en angripare och är ämnat att skada. Detta, menar Six (2011), är viktigt att göra därför att man som utvecklare, inte vet någonting om eller har någon kontroll över datan utanför sin applikation. Han skriver att det finns många sätt att validera input på men nämner två sätt som de mest prominenta. Det första kallar han för reject-known-bad. Med detta menar Six (2011) att man bör, som utvecklare, veta på vilka sätt en angripare kan utnyttja input för att attackera en applikation. Utvecklare bör, utifrån denna kunskap, skapa en applikation som förkastar all input som påminner om denna kända onda input och acceptera all input som inte gör det. Dock så anser han att det andra sättet är ett bättre sätt. Detta kallar han för att accept-known-good, vilket innebär att utvecklare bör skapa applikationer som endast accepterar input baserat på en lista av input som är godtagbar. Utvecklare bör veta vilken typ av input som är godtagbar och säker och endast tillåta denna input i applikationen, all annan input ska förkastas. Det finns ett annat sätt som Miller et al. (2012) nämner där det går att skydda sig från attacker som sker genom input utan att validera inputen. Genom att minimera ytan som är mottaglig för denna typ av attack så minimeras sannolikheten för att en sådan attack lyckas. Detta, menar Miller et al. (2012), görs genom att ha så lite kod som möjligt som är mottaglig för en sådan attack.

Oavsett hur detta implementeras så fortsätter Six (2011) att skriva att validering av input är grunden till säkerheten i mobilapplikationer. Han skriver att mobilapplikationer drivs av input och skadlig input kan driva applikationen till att utföra handlingar med stora negativa konsekvenser. Ett exempel på hur input enkelt kan valideras är att, om man vet att en typ av input inte kan vara över ett visst antal tecken, så går det enkelt att skriva kod som endast tillåter input under ett visst antal tecken. All annan input utöver denna tas inte emot av applikationen. Ingen utvecklare som är kunnig inom säkerhet skulle inte och bör inte lita på input till någon grad. Det finns endast ett tillfälle då man, som utvecklare, kan lita på input och det är när den som skickar input till applikationen, vid ett tidigare tillfälle har autentiserat sig. Om autentiseringen är säker och om det går att lita på att den som skickar input inte har utsatts för någon attack så kan man vara mindre misstänksam mot den data som tas emot (Zdziarski, 2012). Sujithra & Padmavathi (2012) skriver att antalet beslut som applikationen tar baserat på input bör minimeras för att minimera sannolikheten att applikationen manipuleras av en angripare.

Dunham (2008) föreslår dock att utvecklare bör själva åstadkomma input validation attacker för att på så sätt säkerställa att applikationen inte är sårbar för dessa attacker. Det finns olika verktyg på marknaden som författaren föreslår att utvecklare kan använda sig av.

2.2.5 Testa säkerheten

Det är viktigt att inte bara testa funktionaliteten, utan även säkerheten i mobilapplikationer (Dunham, 2008). Att testa säkerheten av mobilapplikationer är både vanligt och nödvändigt om man ämnar skapa en säker applikation (Hoog, 2011). Kadrich (2007) skriver att det är viktigt att testa mobilapplikationer, speciellt om de kommunicerar med en server som innehåller känslig information. Han påminner konstant om hur viktigt det är att testa säkerheten i mobilapplikationer och den största anledningen till detta är, enligt honom, för att säkerhetstester är utvecklarnas sista chans att upptäcka säkerhetsbrister innan någon annan gör det. Kadrich (2007) menar att den som istället upptäcker dessa brister kan ha skadliga avsikter. Han fortsätter med att beskriva att applikationer bör testas på både hög och låg nivå. Gunasekera (2012) skriver att ju mer applikationer testas för säkerhet desto mer säker kan man vara på att applikationen inte är mottaglig för attacker.

Den mest frekventa typen av test som uppkommer i vår litteratur är penetrationstester. Målet med denna typ av tester är att angripa sin egen applikation och se hur den hanterar olika typer av attacker. Efter att man angripit sin applikation så bör de brister som man har upptäckt att applikationen har åtgärdas (Six, 2011). Den information som penetrationstester genererar är extremt värdefull för utvecklare om de ämnar skapa säkra applikationer (Zdziarski, 2012). Man kan lära sig väldigt mycket om sin applikation och hur den fungerar genom att utföra denna typ av tester. Det finns olika verktyg för att testa sin applikation på detta sätt och Dwivedi (2010) föreslår att utvecklare använder sig av dessa, eftersom de sparar tid och energi från att manuellt försöka angripa sin applikation. La Polla, Martinelli & Sgandurrara (2012) skriver att utvecklare enkelt kan hitta fel med hjälp av verktyg som tar lång tid och är svåra att hitta om man istället gör det manuellt. Kadrich (2007) föreslår att de som testar applikationen inte är samma personer som utvecklar applikationen. Detta kan, till stor del, eliminera risken för att slarv uppkommer och skapar negativa konsekvenser för mobilapplikationens säkerhet. Han menar att risken för att vårdslöshet inte upptäcks blir mindre om flera personer ansvarar för att upptäcka fel vid olika tillfällen.

Miller et al. (2012) påpekar att det även är viktigt att ha versioner av applikationen i åtanke. Bara för att en äldre version av applikationen var säker mot en typ av attacker så betyder det inte att den nya versionen är det. Hoog & Strzempka (2011) påpekar att varje applikation bör ha ett minimumkrav för hur mycket den ska testas. De menar att varje applikation som utvecklas inte bör testas mindre än vad detta minimumkrav tillåter. Detta bör göras varje gång en ny applikation utvecklas eller en befintlig ändras. Detta, menar de, kommer både att gynna utvecklare och användare. Ännu en viktig sak som Son, Lee & Oh (2012) nämner angående vad utvecklare bör tänka på när de testar säkerheten i mobilapplikationer är att inte separera testfasen med utvecklingsfasen. De menar att detta kan resultera i att det blir svårt att förstå sig på applikationens svagheter och försöka reparera dessa.

Basavala, Kumar & Agarrwal (2013) skriver att man, utöver att använda sig av verktyg, även kan använda sig av andra företag som underleverantörer, som specialiserar sig på att testa säkerheten i applikationer.

Avslutningsvis så nämner Fried (2010) att säkerhet bör finnas med i utvecklingen av mobilapplikationer. Utvecklare som inte testar sina applikationer kan, enligt honom, inte garantera att applikationen är säker.

2.3 Viktiga säkra arbetssätt

Vi har valt att placera alla frekvent omnämnda faktorer gällande säkra arbetssätt i en kategori. Faktorerna i denna kategori behandlar det som går att hitta i litteraturen gällande hur man bör arbeta för att skapa säkra mobilapplikationer. Dessa påverkar hur utvecklare arbetar med säkerhet ur ett övergripande perspektiv. Dessa faktorer påverkar hela utvecklingen av mobilapplikationer och inte, likt den förra kategorin, specifika delar av utvecklingen.

Som det tydligt går att se i tabell 2 så går det att hitta två ofta förekommande faktorer gällande säkra arbetssätt. Nedan förklaras dessa två faktorer.

2.3.1 Säkra praxiser

För att förhindra att säkerhetsbrister uppkommer från första början så kan utvecklare ha stor användning av praxiser som är utvecklade för att skapa säkra applikationer. Säkra praxiser kan hjälpa utvecklare att inte begå vanliga misstag och slarvfel. Dålig design och implementering förekommer ofta i programmeringsprojekt. Om man har en standard för hur detta ska ske och följer den i varje projekt så underlättar det för utvecklaren (Zdziarski, 2012). Förutom att använda säkra praxiser för att förhindra fel så går det även att använda sig av praxiser för att hantera fel när de uppstår eller för att skapa åtgärder som automatiskt hanterar dessa (La Polla, Martinelli & Sgandurra, 2012). Användningen av säkra programmeringspraxiser skapar även bättre konsistens och gör att det går att leverera olika projekt med liknande kvalitet. Det finns många olika praxiser för utvecklare att följa, men de som har visat sig vara mest gynnsamma för säkerhet har varit agila metoder. Agila metoder brukar resultera i att utvecklare gör färre fel jämfört med andra sorters metoder. Anledningen kan vara att de ofta återkommer till samma problem vid flera tillfällen (Six, 2011).

Kadrich (2007) anser att en standard för hur man programmerar som är säker är en nödvändighet om man ämnar utveckla säkra applikationer. Dålig kod skapar svagheter som angripare kan dra nytta av. Ramu (2012) nämner ett exempel om utvecklare som inte följde säkra praxiser och hade som vana att placera känsliga detaljer på platser som gjorde det enkelt

för angripare att komma åt dessa. Förutom att praxiser minimerar risken för slarvfel så gör de även att utvecklare slipper ta nya beslut vid varje projekt, besluten för hur de bör programmera har då redan etablerats och finns i praxiserna och kan på så sätt återanvändas (Kadrich, 2007).

Dessa praxiser består ofta av best practices, som exempelvis att kod alltid bör signeras (Dunham, 2008). Ett annat exempel är, som tidigare nämnts, att minska andelen kod som är mottaglig för en input validation attack (Miller et al., 2012). Ytterligare ett exempel på best practices är att det finns standarder för hur man bör ta sig till när man krypterar (Gunasekera, 2012). Det finns många saker som utvecklare bör göra men inte gör. Säkra praxiser bör ha med sådana saker (Dunham, 2008). Utvecklare bör ha i åtanke att kända praxiser oftast bygger på många år av undersökningar och forskning, vilket gör att utvecklare kan dra nytta av andra mer kunniga personers erfarenhet istället för att själv försöka skapa säkra riktlinjer. Eftersom projekt där mobilapplikationer utvecklas ofta har en kort deadline och eftersom utvecklare i dessa projekt ofta finner sig själva i stressande situationer så kan riktlinjer för hur utvecklare snabbt och enkelt skapar säkra applikationer på bästa sätt vara till stor hjälp (Dwivedi, 2010).

Utöver att säkra praxiser som redan finns används så kan man även forma sina egna praxiser med hjälp av egen erfarenhet. Om tidigare arbetssätt som har fungerat på ett bra sätt vid flera tillfällen används så går det att skräddarsy sina säkra praxiser efter de egna behoven. Dock måste man ha i åtanke att den egna erfarenheten inte är lika stor som erfarenheten bakom de redan existerande och kända säkra praxiserna (Hoog, 2011).

Oavsett vilka praxiser som används så bör alla som utvecklar mobilapplikationer ha en standard säkerhet som är minimumkravet för alla applikationer som utvecklas. På så sätt minskas sårbarheten hos alla ens applikationer, oavsett hur de utvecklas (Fried, 2010). Som det tidigare nämnts så skriver Son, Lee & Oh (2012) att säkra praxiser inte bör separera säkerhetstester och utveckling av säkerhet eftersom det kan resultera i att det blir svårare att förstå problemen samt hantera dem.

Avslutningsvis så påpekar Hoog & Strzempka (2011) att utvecklare, inom en snar framtid, kommer vara tvungna att använda sig av säkra praxiser för att kunna skydda användarna av applikationer som de utvecklar.

2.3.2 Användning av verktyg

Innan en utvecklare förklarar en mobilapplikation att vara helt säker så måste man erkänna att det kan finnas en möjlighet till att den kan innehålla vissa säkerhetsbrister (Zdziarski, 2012). Som det tidigare nämnts så kan säkerheten i en applikation testas med hjälp av olika verktyg (Dwivedi, 2010). Dessa verktyg kan kontrollera en applikations säkerhet på ett snabbare och bättre sätt än vad utvecklare kan. Dessa verktyg hjälper till genom att de försöker skapa så mycket skada de kan på så många olika sätt de kan. Man observerar sin egen applikations reaktion på dessa attacker och reparerar den utefter den information man samlar på sig med hjälp av dessa verktyg. Exempel på attacker som kan utföras för att testa säkerheten är kapning av SSL sessioner och brute force attacker. Eftersom angripare ofta använder sig av verktyg så kan utvecklare också göra detta, i ett proaktivt syfte (Zdziarski, 2012). Dock så är det viktigt att utvecklare använder sig av verktygen på rätt sätt. Om utvecklare förlitar sig på att ett verktyg ansvarar för en del av applikationens säkerhet så måste utvecklare säkerställa att de förstår verktyget och använder det rätt (Hoog & Strzempka, 2011).

Verktyg kan även användas på andra sätt. Man kan, exempelvis använda verktyg för att på ett snabbare och enklare sätt kryptera och skydda data. Man kan även använda verktyg för att obfuskera sin källkod och på sätt göra det mer komplicerat för en angripare att ge sig på applikationen (Six, 2011). Ytterligare hjälp som verktyg kan erbjuda är att de kan ändra tillåtelser på ett snabbt sätt. De kan exempelvis ta bort rättigheter att avläsa viss data för flera entiter på ett snabbt sätt. De kan enkelt samla och visa information, exempelvis om nätverket, som kan hjälpa utvecklare att ta beslut om hur applikationen bör se ut. Man kan även testa nätverksanslutning och överblicka transporten av data (Kadrich, 2007). Det finns även verktyg som kan hjälpa utvecklaren att testa om applikationen uppnår en grundläggande säkerhetsnivå (Hoog, 2011) eller underlätta processen för reverse engineering (Gunasekera, 2012). Verktyg kan även, utan att utsätta applikationen för attacker, analysera koden efter kända svagheter (Son, Lee & Oh, 2012). Ett sista exempel på hur verktyg kan hjälpa utvecklare är att de kan extrahera en viss typ kod som manuellt hade tagit lång tid att extrahera (Miller et al., 2012).

Son, Lee & Oh (2012) skriver att alla utvecklare inte är kunniga inom säkerhet och att denna kunskap tar lång tid att anskaffa sig. På grund av detta anser de att verktyg kan vara till hjälp för utvecklare med mindre än önskvärd kunskap inom säkerhet.

Den viktiga punkten när det gäller verktyg är att användningen av dem kan förenkla och påskynda utvecklingen av mobilapplikationer med kvalitativ säkerhet. En stor del av jobbet slipper utvecklare göra och en stor del av jobbet sker snabbare än vad utvecklare själva kan göra (Zdziarski, 2012). Hoog (2011) påpekar att det är viktigt att använda den senaste versionen av sina verktyg. Gamla versioner kan inte garantera total säkerhet och eftersom utvecklare ibland förlitar sig på vissa verktyg gällande säkerheten så bör de garantera att dessa verktyg utför sitt jobb på bästa sätt.

2.4 Säker kultur

Vi har valt att placera alla frekvent omnämnda faktorer gällande säker kultur i en kategori. Faktorerna i denna kategori behandlar det som går att hitta i litteraturen gällande hur utvecklare bör ta sig till för att skapa en kultur som gynnar utveckling av säkra mobilapplikationer.

Som det tydligt går att se i tabell 3 så är den första punkten, kunskap, väldigt frekvent omnämnd. Dock är den andra faktorn, säkerhetsmål, mindre omnämnd. Trots detta har vi valt att inkludera denna faktor. Detta har vi gjort för att vi anser att antalet källor som skriver om denna faktor är hög nog för att den ska inkluderas i vårt ramverk.

2.4.1 Kunskap

Kunskap är viktigt för utvecklare när det gäller säkerhet (Gunasekera, 2012). För att utvecklare ska kunna utveckla säkra applikationer så måste de vara kunniga inom säkerhet. Utvecklare utan kunskap inom säkerhet kan inte utveckla applikationer lika säkert som utvecklare som har denna kunskap (Kadrich, 2007). Zdziarski (2012) anser att utvecklare som inte är kunniga inom säkerhet kommer att göra fel. Six (2011) understryker vikten av utbildning och anser att det krävs mycket utbildning för att skapa säkra applikationer. Utbildning inom säkerhet ökar kunskapen och pålitligheten för utvecklare att skapa säkra applikationer, men det gör även att utvecklare blir mer säkerhetsorienterade. Han menar att utvecklare tänker mer på säkerhet och blir mer medvetna om säkerheten ju mer de utbildas

inom den. Hoog & Strzempka (2011) skriver att många utvecklare applicerar säkerhet endast för att det är ett krav, men de har ingen större förståelse för hur viktig säkerheten är för användarna av applikationen.

Zdziarski (2012) påpekar att angripare oftast har mer kunskap om säkerhet än utvecklare, vilket innebär att de ofta kan göra saker som utvecklare inte vet att man kan göra. De kan angripa på sätt som utvecklare inte är medvetna om. Han exemplifierar detta genom att skriva om hur utvecklare använder och litar på SSL utan att veta på vilket sätt det kan fallera. På grund av detta så anser Zdziarski (2012) att utvecklare bör utbildas i säkerhet. Han anser, likt Kadrich (2007) och Six (2011), att ju mer utvecklare vet, desto säkrare blir applikationerna.

Dwivedi (2010) nämner frekvent att utvecklare bör vara medvetna om vilka hot som finns och hur man bör skydda sig mot dessa. Han menar att angripare gynnas av att utvecklare inte är säkerhetsmedvetna. Dock menar Son, Lee & Oh (2012) att verktyg, som tidigare nämnts, kan vara till hjälp för utvecklare som inte har mycket kunskap när det gäller säkerhet.

Ju mer kunskap utvecklare har desto mer avancerade saker kan utföras, vilket resulterar i att man är bättre rustad för att hantera flera olika typer av hot. Det blir då enklare att anpassa sig efter de krav som ställs gällande säkerhet (Miller et al., 2012). Fried (2010) fortsätter på detta spår genom att skriva att det är viktigt för utvecklare att anpassa sig efter den senaste tekniken och inte ligga kvar i gammal kunskap.

2.4.2 Säkerhetsmål

Fried (2010) skriver att säkerhet i grunden handlar om att vara medveten om vilka risker applikationen ställs inför. Det är av största vikt att utvecklare är medvetna om vilka risker som finns och hur man som utvecklare bör förhindra dessa. Man bör alltid börja med att skapa säkerhetsmål som bör uppnås vid utveckling av applikationer. Alla som deltar i projektet bör vara medvetna om dessa mål. De hjälper utvecklarna att vara medveten om de risker som kan uppstå samtidigt som vissa säkerhetsåtgärder inte glöms. Riskerna bör utvärderas efter hur känslig datan är och efter hur sannolikt det är att risken inträffar. I varje projekt så finns en önskad säkerhetsnivå och om inte denna nivå uppnås så kan det medföra allvarliga negativa konsekvenser. Att inte vara medveten om ens säkerhetsnivå är någonting som borde undvikas mycket. Det finns många sätt att definiera dessa säkerhetsmål på, ett exempel är att ha en lista över dem och ett annat är att skapa användarfall där säkerheten testas (Fried, 2010). Ytterligare ett exempel på hur man kan definiera säkerhetsmål är att använda sig av modellering (Kadrich, 2007).

Att modellera hot hjälper utvecklare att förstå hur en angripare kan attackera applikationen genom att visa var känslig data finns, var den är på väg, vem som kan komma åt den och hur den existerar på just den platsen. Man kan modellera på två olika huvudsakliga sätt. Det första är att endast modellera övergripande och inte ha med detaljerad information och det andra sättet är att modellera väldigt detaljerat där man försöker förstå applikationen på djupet. Modellering kan även hjälpa alla medverkande i ett projekt att förstå applikationen och dess säkerhet på samma sätt (Kadrich, 2007). Även Basavala, Kumar & Agarrwal (2013) nämner att modellering är en bra metod för utvecklare att använda sig av när de utvecklar säkerhet. De skriver att en modell bör ha med fem saker. Den första är arkitekturen av den mobila applikationen, den andra är datan i applikationen, den tredje är identifiering av angripare, den fjärde är metoder av attacker och den femte är åtgärder för att skydda sig mot attacker. La Polla, Martinelli & Sgandurrara (2012) skriver att om utvecklare modellerar de konsekvenser

som kan ske ifall mobilapplikationen blir attackerad kan vara till stor fördel för dem när de arbetar med säkerhet. Ramu (2012) föreslår att man bör modellera efter angriparens perspektiv, vilket innebär att man skapar en modell över angriparens möjligheter och begränsningar, istället för att modellera efter ett defensivt perspektiv.

2.5 Sammanfattning

I vår analys av vår litteratur så har vi funnit nio frekvent återkommande faktorer som bör finnas med i utveckling av säkra mobilapplikationer. Dessa faktorer är kryptering av kommunikation, autentisering, säker lagring av data, validera input, testa säkerheten, säkra praxiser, användning av verktyg, säkerhetsmål och kunskap. Faktorerna har vi sedan delat upp i tre kategorier. Dessa kategorier är tekniska åtgärder, säkra arbetssätt och säker kultur. Alla faktorer utgör tillsammans det teoretiska ramverk som vi kommer att använda oss av i vår studie för att svara på vår forskningsfråga.

3 Metod

3.1 Tillvägagångssätt

Fokus i denna uppsats har varit att först och främst genom litteraturen identifiera och presentera faktorer som företag bör utgå från vid utveckling av säkra mobilapplikationer. Genom att tillskansa oss litteraturen så kunde vi sammanställa faktorer som av litteraturen ansågs vara viktiga. Denna sammanställning möjliggjorde för oss att göra en kartläggning av de faktorer som nämndes mest frekvent i litteraturen. Vi identifierade totalt nio faktorer som sedan delades in i tre kategorier. Dessa faktorer redovisades med hjälp av tabeller och samt beskrivningar i vår litteraturgenomgång. Detta utgjorde i sin tur vårt teoretiska ramverk. De tre kategorierna med dem nio faktorerna är:

1. *Tekniska åtgärder*
Kryptering av kommunikation, autentisering, säker lagring av data, validera input, testa säkerheten.

2. *Säkra arbetssätt*
Säkra praxiser, användning av verktyg.

3. *Säker kultur*
Säkerhetsmål, Kunskap

Vårt teoretiska ramverk användes sedan som grund till att utforma vårt intervjuformulär. Detta gjordes för att kunna tillämpa insamlad empirisk data på vårt teoretiska ramverk.

3.2 Val av undersökningsmetod

Eftersom vår frågeställning är av explorativ karaktär så lämpar sig en kvalitativ undersökning bäst enligt Jacobsen (2002). Vi kände att en kvalitativ undersökning gav oss möjligheten till att gå in mer på djupet samt att vi fick fram en mer mångfacetterad data. En kvalitativ metodansats är enligt Jacobsen (2002) att föredra när man vill se ett samband mellan individ och kontext. I vårt fall blev individen respondenten vi intervjuade medan kontexten blev hur utvecklare inom företaget inför säkerhet när det gäller utveckling av mobilapplikationer. Sambandet mellan respondent och företag kan i vårt fall liknas vid att en anställd som utvecklar mobilapplikationer är en representant för företaget och dennes handlingar kring säkerhet avspeglar och utgår från ett företags uppsatta riktlinjer.

3.3 Datainsamling

Vid genomförandet av denna undersökning intervjuade vi fyra stycken IT-konsultföretag. I enlighet med Trosts (2010) synsätt vid arbete med empiriska undersökningar delades arbetet upp i tre faser; insamling av data genom ett antal intervjuer, analys, samt tolkning av datan. Vi valde att göra semistrukturerade intervjuer då dessa gav oss möjligheten för en mer öppna och djupare diskussioner. Intervjuerna skedde på plats hos våra respondenter i samtliga fall, detta medförde att det skapades en mer lättsam stämning. För att vi fullständigt skulle koncentrera oss på respondenterna och kunna komma med följdfrågor som är en del av den semistrukturerade intervjun så använde vi oss av en portabel ljudinspelare. Eftersom vi hade detta verktyg till vår hjälp så fördes inga anteckningar under intervjuerna, vilket i sin tur ledde

till att vi blev beroende av utrustningen. För att säkerställa att detta beroende inte skulle utgöra ett problem för oss under intervjuerna så hade vi två separata ljudinspelare.

3.4 Urval av respondenter

Undersökningen utfördes på fyra olika företag. Samtliga företag är IT-konsultföretag som, bland annat, utvecklar mobilapplikationer. Samtliga respondenter medverkar i utvecklingen av mobilapplikationer, är kunniga inom säkerhet och vet både hur företaget utvecklar mobilapplikationer och säkerheten för dessa. Företagen var medvetna om att de personer som intervjuades var representanter för företaget inom deras geografiska område, vilket var viktigt för att vår respondent skulle vara så representativ för företaget som möjligt.

Namn	**Företag**	**Tjänst**
Johan Enell	Cybercom	IT-utvecklare
Petter Sandholt	Softhouse	Utvecklare och säkerhetskonsult
Anonym	Anonym	Affärschef
Björn Nilsson	Xdin	Utvecklare

Tabell 4: Översikt av respondenter

För att respondenterna skulle kunna intervjuas i vår studie så hade vi sex kriterier som varje respondent var tvungen att uppfylla. Följande kriterier hade vi på våra de respondenter:

- Företaget är ett IT-konsultföretag
- Företaget utvecklar mobilapplikationer
- Respondenten medverkar i utveckling av mobilapplikationer
- Respondenten är kunnig inom säkerhet
- Respondenten vet hur företaget utvecklar mobilapplikationer
- Respondenten vet hur företaget utvecklar säkerhet i mobilapplikationer

3.5 Utformning av intervjuguide

Syftet med våra intervjufrågor var att få fram svar som beskrev om företagen inkluderar de faktorer som vår litteratur tar upp och hur deras arbete med dessa ser ut. För att få så bra svar som möjligt så valde vi att strukturera frågorna efter vårt teoretiska ramverk. Intervjufrågorna har underfrågor som var ämnade att få fram detaljerad information gällande hur de arbetar, medan huvudfrågan till varje fråga endast var till för att ta reda på om det arbetar på ett visst sätt. Vi inledde intervjun med två frågor som inte är formade efter vårt teoretiska ramverk. Den första frågan var ämnad att få fram information gällande respondentens erfarenhet och kunskaper. Den andra frågan var övergripande och dess syfte var att få svar på de resterande frågorna innan de ställdes, för att minimera vår påverkan av respondenternas svar och för att upptäcka om företagen arbetade på ett sätt som inte tas upp av vår litteratur. Vår avslutande fråga ställde vi av samma anledning som vår andra fråga, den var ämnad att ta reda på om företaget arbetade med säkerhet på ett sätt som inte tas upp i våra intervjufrågor. Alla frågor från fråga tre till fråga 14 var direkt kopplade till vårt teoretiska ramverk. Våra frågor var följande:

1. Skulle du kunna berätta lite om dig själv och din bakgrund?

2. Skulle du kunna beskriva hur ni tar hänsyn till säkerhet vid utvecklingen av era mobilapplikationer?

3. Krypterar ni kommunikationen till och från era mobilapplikationer?
    a. Hur tar ni er till när ni krypterar denna kommunikation?

4. Validerar ni input i era mobilapplikationer?
    a. Hur tar ni er till när ni validerar input?

5. Använder ni er av autentisering i era mobilapplikationer?
    a. Hur tar ni er till när ni autentiserar?

6. Testar ni säkerheten i era mobilapplikationer?
    a. Hur tar ni er till när ni testar säkerheten?

7. Använder ni er av verktyg för att förbättra säkerheten i era mobilapplikationer?
    a. Vilka verktyg använder ni er av?

8. Har ni några metodologier eller riktlinjer som ni följer under utvecklingen av era mobilapplikationer?
    a. Vilka är dessa?
    b. Hur tror ni att dessa hjälper er med säkerheten?

9. Definierar ni säkerhetsmål i början av era projekt?
    a. Hur går ni till väga när ni definierar säkerhetsmål?

10. Hur går ni till väga för att lagra data säkert i era mobilapplikationer?

11. Kartlägger eller modellerar ni hoten i början av era projekt?
    a. Hur tar ni er till när ni gör detta?

12. Erbjuder ni era anställda utbildningar inom säkerhet?
    a. Vilken utbildning erbjuder ni era anställda inom säkerhet?

13. Håller ni er uppdaterade om vilka hot och risker som existerar?
    a. Hur tar ni er till när ni gör detta?

14. Håller ni era utvecklingsverktyg uppdaterade?

15. Finns det någonting som du vill tillägga?

Dessa frågor var utformade på detta sätt för att skulle kunna analysera våra empiriska data efter samma struktur som vårt teoretiska ramverk, vilket skapar en konsistens i uppsatsen och gör den enklare att läsa och överblicka.

3.6 Bearbetning av empiri

Genomförandet av våra fyra intervjuer resulterade i mycket rådata. Jacobsen (2002) menar att det är viktigt att fråga sig själv, hur pass komplett ens registrering av data är. Han påpekar

också att den mest fullständiga registreringen får man när någon form av inspelningsanordning används och att rådata som fås med hjälp av dessa hjälpmedel därför är idealet vid kvalitativa metoder. Vid utförandet av våra intervjuer valde vi därför att använda oss av en digital ljudinspelare samt för att vara på den säkra sidan hade vi ytterligare en ljudinspelare, då vi ville skydda oss mot eventuella tekniska problem som skulle kunna dyka upp. Detta resulterade även i att vi kunde fokusera oss helt på frågorna och svaren under intervjuernas gång, då vi slapp föra anteckningar. Jacobsen (2002) säger samtidigt att en konsekvens av detta är att det resulterar i att man får mycket rådata som måste hanteras. Detta är i linje med vad Trost (2010) också påpekar, det vill säga att när mycket rådata ska transkriberas så blir det besvärligt att göra detta på ett korrekt sätt. Av tidigare erfarenhet i andra studier så var vi införstådda med detta och vi visste att det både skulle vara tidskrävande och påfrestande när vi valde att göra en fullständig transkribering på alla intervjuer.

Detta gjorde i sin tur det enklare för oss i efterhand då vi kunde sammanfatta intervjuerna för användning i vår analys. Strukturen på analysen byggdes upp på samma sätt som vår intervjuguide. Detta gjorde vi för att säkerställa att ett konsekvent upplägg genomsyrade hela vår uppsats.

3.7 Källkritik

Enligt Trost (2010) så är det viktigt att ha ett kritiskt förhållningssätt när data samlas in. Med detta menar han att det är viktigt att man har klart för sig var informationen kommer ifrån samt hur pass pålitlig källan är. Vi valde därför att arbeta efter Thuréns (2005) fyra principer när det gäller källkritik. Dessa principer är:

- Äkthet: Källan måste vara sann och oförfalskad.

- Tidssamband: Källan måste vara relevant ur ett tidsperspektiv, det vill säga att man bör sträva efter att källorna ska vara så nya som det möjligen går för det man undersöker. På så sätt så minskas skälen till att tvivla på källan.

- Oberoende: Källan bör vara oberoende av andra källor, det vill säga att den inte ska vara ett referat av en annan källa.

- Tendensfrihet: Misstänksamhet för om källan verkligen berättar sanningen om verkligheten på grund av ekonomiska, personliga och politiska intressen ska inte finnas.

Äkthet

Böckerna som vi använt oss av har alla kommit från välrenommerade författare med mycket erfarenhet. När det gäller våra akademiska artiklar så har vi bara använt oss av LUBsearch samt Google Scholar. I princip alla våra källor behandlar bara mobilapplikationer.

Eftersom alla våra respondenter är kopplade till just utveckling av mobilapplikationer på sina respektive arbetsplatser så vet vi att de är vilka de utger sig för att vara.

Tidssamband
När det gäller vår litteratur så har vi valt att använda oss av så nya källor som möjligt, en stor majoritet av våra källor är mindre än två år gamla.

När det gäller vår empiri så arbetar våra respondenter dagligen med utveckling av mobilapplikationer och är därmed införstådda gällande den nyaste informationen på deras arbetsplatser när det gäller säkerhet vid utveckling av mobilapplikationer.

Oberoende
När det gäller litteraturen så har detta steg följts genom att vi granskade både våra böcker samt akademiska artiklar för att säkerställa att dessa källor är oberoende.

Eftersom våra respondenter hade kunskapen att svara på alla våra frågor direkt så säkerställdes oberoende, och var det något de var osäkra på så fick vi reda på det.

Tendensfrihet
När det gäller litteratur har just användningen av både böcker samt akademiska artiklar möjliggjort för oss att verkligen se till att källan är sanningsenlig.

Våra respondenters svar uppfyllde denna princip genom att de inte hade någon anledning att inte berätta sanningen för att vinna något på det.

3.8 Kritik av metodval
Ett stort problem med den kvalitativa metoden enligt Jacobsen (2002) är att den är resurskrävande och på grund av detta får man många gånger nöja sig med ett fåtal respondenter. Detta leder i sin tur till att undersökningen inte är statistiskt representativ. Eftersom vår empiriska undersökning ägde rum hos fyra företag så kan vi inte dra en slutsats som är representativt för mer än dessa fyra företag.

3.9 Validitet och realibilitet
Enligt Jacobsen (2002) så bör empirin, oavsett typ, uppfylla två krav:

1. Empirin måste vara giltig och relevant *(valid)*. Med detta menar Jacobsen (2002) att man verkligen mäter det vad man ämnar mäta, samt att de uppskattningar som gjorts hos ett fåtal är detsamma för ett flertal, det vill säga att man ska kunna föra över resultatet från en kontext till ett annat. Vill man uppfylla detta bör det som undersökts vara karakteristiskt för det kontext som vi vill överföra det till (Jacobsen, 2002).

För att kunna uppnå hög validitet så har vår intervjuguide varit utformat på så sätt att när svaren analyserats så har vår intention varit att vi ska svara på vår frågeställning. Vi har också vid kontakt med företagen påpekat vad exakt vår uppsats kommer att behandla för att företagen på så sätt ska kunna frigöra en respondent som har kunskapen om att svara på våra frågor och därmed kunna öka vår validitet.

2. Empirin måste vara tillförlitlig och trovärdig *(reliabel)*. Här påpekar Jacobsen (2002) att undersökningen måste vara pålitlig. En kontrollfråga man kan ställa sig kan vara om man hade fått samma resultat ifall samma samma undersökning fullföljts ännu en gång? När en

undersökning genomförts på exakt samma sätt och givit samma resultat, kan man säga att undersökningen har uppnått hög tillförlitlighet (Jacobsen, 2002).

För att uppnå hög reliabilitet så har vi under intervjuerna använt oss av ljudinspelare för att säkerställa att all information som vi fått är tillförlitlig och trovärdig. Analysen genomfördes utifrån den fullständiga transkriberingen, samt sammanfattningen av den. Vi bad också våra respondenter ifall vi kunde höra av oss igen ifall det var något som var oklart för oss under den fullständiga transkriberingen för att isåfall kunna ändra något som blivit fel. Vi tror att vi hade uppnått samma resultat om vi hade utfört samma undersökning igen, då respondenterna hade olika års erfarenhet samt att alla företagen verkar inom samma bransch.

3.10 Etik

Enligt Jacobsen (2002) så finns det tre aspekter som man bör sträva efter och förhålla sig till när det gäller etik i en undersökning. Dessa är: *informerat samtycke, rätten till ett privatliv och vikten av korrekt data.*

Informerat samtycke innebär att respondenten ska känna att denne ställer upp frivilligt, detta beslut ska enligt Jacobsen (2002) ha grundats sig på övervägande av de möjliga riskerna och vinsterna som medföljer vid ett deltagande i en undersökning. På grund av detta var det viktigt för oss att belysa respondenterna om vad målen med vår undersökning var samt att de gärna kunde ta del av den när den var klar. Vi tog heller aldrig för givet eller påtvingande våra respondenter användningen av ljudinspelare, utan vi frågade ifall det gick bra att göra det.

Enligt Jacobsen (2002) innehåller den andra aspekten, rätten till ett privatliv tre underkategorier. Den första underkategorin är att man ska överväga känsligheten på informationen som man vill samla in. Den andra underkategorin man bör ha klart för sig är hur känslig informationen är och den sista underkategorin handlar om deltagaren kan identifieras utifrån den insamlade datan. Eftersom vår undersökning behandlar säkerhet så visste vi att säkerhet alltid är ett känsligt ämne, därför så var bland det första vi påpekade för deltagaren att om någon av våra frågor leder in till ett svar som var kopplad till eller ansågs vara en företagshemlighet, så hade deltagaren möjligheten och rätten att undvika frågan. Detta var vi noga med för att vi bestämde oss för att i början fråga det gick bra om namnet på företaget samt på deltagaren kunde stå med i vår undersökning. Detta gav inte mycket mervärde för vår undersökning men anledningen till att vi valde att göra så var för att vi, för det första; ansåg att vår uppsats blir mer läsbar, för det andra; att vi var väldigt tacksamma för de företag- och anställda som valde att tacka ja till vår uppsats och på grund av detta ville vi uppmärksamma dem genom att ha deras namn i vår uppsats.

Den sista aspekten som Jacobsen (2002) tar upp beträffande etik är vikten av korrekt data. Med detta menar Jacobsen (2002) att information inte ska tas ur sitt sammanhang och att data och resultat inte heller ska förfalskas, därför är det viktigt att resultatet ska, i största möjliga utsträckning, återges korrekt. Denna aspekt var något som vi tog mycket hänsyn till då vi från början bestämde för att vi helst ville spela in intervjuerna så att informationen skulle återges så korrekt som möjligt var vi tvungna att fråga om lov om det gick bra att vi spelade in deltagarna. Vi avslutade också intervjuerna med att fråga ifall vi kunde återkomma om vi kände att någonting var oklart under transkriberingen.

4 Empiriska resultat

I detta kapitel presenteras företagen som vi har intervjuat och resultatet av vår empiriska undersökning. Kapitlet är strukturerat efter vårt teoretiska ramverk, där rubrikerna följer de faktorer som tas upp i ramverket. I slutet av kapitlet sammanfattas resultatet av den empiriska undersökningen.

4.1 Presentation av företag

Samtliga företag i vår studie är IT-konsultföretag som utvecklar mobilapplikationer. Alla företag utvecklar inte enbart mobilapplikationer, utan även andra former av mjukvara.

Cybercom erbjuder tjänster inom kommunikation så som digital solutions, connected engineering, connectivity management och secure connectivity. Företaget grundades år 1995, det har 1335 anställda och finns för det mesta i Norden.

Softhouse grundades år 1996 och har omkring 100 anställda. Företaget finns i Sverige, Finland och Indien och är ledande inom Lean Software Development i Skandinavien. Softhouse erbjuder, utöver mjukvaruutveckling, även agil coaching och utbildning.

Anonym finns internationellt, men Norden är dess utgångspunkt. Företaget grundades 1986 och har över 1000 anställda. Anonyma siktar på långsiktiga affärer där deras kunskaper inom informationsteknologi ska gynna deras kunders verksamhet.

Xdin grundades 1991 och har omkring 1200 anställda. Företaget existerar i Sverige, Norge och USA med Sverige som utgångspunkt. Xdins största verksamheter är beräkning och simulering, produktutveckling, IT- och systemutveckling, elektronik- och mjukvaruutveckling, utbildning, support och metodutveckling.

4.2 Övergripande säkerhetsåtgärder

Cybercom anpassar säkerheten i sina applikationer utifrån två faktorer. Den första är tid och den andra är datan som applikationen hanterar. Brist på tid medför att säkerheten bortprioriteras. Säkerhet bortprioriteras även om datan som applikationen hanterar inte anses vara känslig nog. Cybercom anser att applikationer inte kan vara fullständigt säkra och har som regel att skapa applikationer med säkerhet som gör att ansträngningen, för angripare, att komma åt datan inte motsvarar värdet av datan som de försöker komma åt. För att säkerställa att applikationerna är säkra har Cybercom ett säkerhetsteam i Stockholm som ansvarar för att applikationen har nått den nivån av säkerhet som det är bestämt att applikationen ska ha.

Softhouse anpassar säkerheten efter vad kunden efterfrågar och vanligtvis efterfrågas säkerhet inte till stor del. Kunder vill ofta ha funktionalitet, användbarhet och design. Säkerhet brukar ofta försämra användarvänligheten och för mycket säkerhet kan göra så att ingen vill använda det som har utvecklats. Softhouse försöker få in säkerhet i sina projekt genom att göra sig medveten om vilka risker som existerar och presentera dessa för kunden, men om kunden inte är intresserad av säkerhet så accepterar Softhouse deras beslut. I sin riskbedömning så analyserar Softhouse sannolikheten för att det händer samt skadan som sker ifall det händer.

Anonym utgår alltid efter kundens behov. De tar alltid hänsyn till vilken applikation de ska utveckla. De har, i alla projekt, minsta möjliga säkerhetsnivåer som de följer. De arbetar oftast med mobilintegration där deras mobilapplikationer kommunicerar med företagssystem.

Xdin förlitar sig mycket på sina utvecklare, de litar på deras egna sunda förnuft. De anpassar de sig mycket efter sina kunder och vad de kräver. Om kunder inte vill betala för säkerhet så spenderar inte Xdin tid på säkerhet. Dock brukar de applicera viss säkerhet även om kunder inte har efterfrågat säkerhet, vilket i deras ord innebär att undvika att göra dumma saker bara för att kunden inte vill ha säkerhet.

4.3 Kryptering av kommunikation

Cybercom använder sig av färdiga standarder när de krypterar kommunikation. HTTPS är ett exempel på en färdig standard som Cybercom använder sig av. När de lagrar data som sparas på enheten som applikationen befinner sig i så använder de sig av de färdiga standardramverken som redan existerar i den miljö de utvecklar för. När de, exempelvis, utvecklar för Android så använder de sig av Javas egna krypteringsbibliotek.

Softhouse krypterar kommunikation efter vad de tycker är rimligt. Om de ska hämta information vill de skydda sig från mannen-i-mitten attacker, men om informationen de hämtar är information om väder så tycker de att de inte behöver skydda sig från den typen av attacker. Krypteringen av kommunikation styrs även av kunderna om Softhouse inte också utvecklar back-end för projektet. Kundernas egna back-end styr hur mycket och på vilket sätt det går att kryptera information som skickas. Kommunikation brukar vanligtvis krypteras mellan applikation och server.

Anonym krypterar kommunikationen med hjälp av HTTPS och SSL. De anser att detta inte är tillräckligt för att säkerställa att kommunikationen är säker. På grund av detta så brukar de även kryptera data själva innan de skickar den vidare. När de själva krypterar använder de sig av AES-kryptering, vilket står för Advanced Encryption Standard. De anser att data inte får existera i server om den inte är krypterad.

Xdin brukar använda sig av HTTPS som kryptering vid kommunikation, dock bestämmer projektet om kryptering krävs vid kommunikation. Om datan inte är känslig så behövs ingen kryptering. Xdin försöker kryptera så mycket som möjligt men försöker samtidigt göra så att krypteringen inte utgör för stor belastning.

4.4 Autentisering

Endast enkel autentisering sker i applikationen hos Cybercom, som exempelvis att se till att inloggningsuppgifterna är korrekta. Resten av autentiseringen sker i back-end.

Softhouse är noga med att kontrollera att certifikaten från de källor som ämnar kommunicera med applikationen är korrekta. De är försiktiga med att lita på certifikat. Softhouse använder sig bland annat av SSL när de krypterar. Vanligtvis använder de sig av kända certifikat, som finns i de flesta mobiltelefoner för autentisering men när projektet kräver det kan de skapa egna certifikat. Även när det gäller autentisering så styr projektet mycket. Om applikationen som utvecklas hanterar en stor del känslig data så försöker de införa autentisering genom att rekommendera detta för kunden. Om applikationen inte hanterar känslig data så behövs ingen autentisering.

Anonym använder sig av device authentication och user authentication när de autentiserar. För att autentisera så tar de hjälp av de autentiseringar som redan existerar i .NETs ramverk.

De använder sig även av SSL men anser att detta inte är säkert nog, i vilket fall de, som det tidigare nämnts, själva krypterar datan.

Xdin autentiserar sina applikationer utifrån vad utvecklaren tycker är rimligt. Om utvecklaren följer ett ramverk eller några riktlinjer beror på från projekt till projekt, om kunden efterfrågar det eller inte. De brukar använda sig av signaturer för att säkerställa att de entiteter som kommunicerar med applikation är de som de utger sig för att vara. De brukar signera i koden och i datan som skickas.

4.5 Säker lagring av data

När Cybercom lagrar data på applikationen så använder de sig av plattformens egna kontexter. Cybercom anser att plattformarnas egna inbyggda funktioner för säkerhet är enklast att använda sig av då det blir för mycket arbete att utveckla egna funktioner.

Softhouse krypterar inte data i applikationen. De placerar data i secured storage som redan finns färdigt på plattformen de utvecklar för. De använder sig inte av någon annan form av säker datalagring i applikationen. De lagrar även data utanför secured storage, denna datan är alltid oviktig data som de anser inte behöver lagras på ett säkert sätt.

Anonym krypterar data som lagras i applikationen om de anser att datan är känslig nog för att kräva kryptering. Anonym använder sig inte av den säkra datalagringen som redan finns i plattformarna. De anser att dessa är det första angripare attackerar och vill inte ha känslig data där. De lagrar istället datan på ett eget säkert sätt och försöker säkerställa att all data i applikationen alltid är krypterad.

Xdin förlitar sig på det som redan finns inbyggt i plattformen när det gäller säker datalagring.

4.6 Validera input

Cybercom gör endast enkel validering av input i applikationen. Applikationen utför endast enkla kontroller som att en email verkligen är en email och att ett telefonnummer verkligen är ett telefonnummer. Det stora ansvaret för att se till att input inte är skadlig lägger Cybercom på back-end servern och inte på applikationen.

När det gäller validering av input så litar Softhouse på sina utvecklare att de tar goda beslut. Utvecklarnas beslut ska grundas på den utbildning som Softhouse erbjuder. De ska förstå vilka risker som existerar om input inte valideras, men Softhouse utför inga kontroller på att input har validerats. Dock valideras inte input ifall applikationen endast fungerar som ett skal, det vill säga att känslig information inte kan nås via applikationen. En vanlig typ av validering de gör är rimlighetsvalidering då de kontrollerar att input är rimlig. Om input exempelvis är lång, då kort input förväntas så kontrolleras detta. Softhouse försöker på grund av detta använda sig av encoding för att se till att datan kan skickas vidare utan problem. De litar mycket på back-end när det gäller validering av input.

Anonym validerar all input i sina applikationer.

Xdin validerar input utifrån vad utvecklaren själv känner är rimligt. Hur mycket som valideras beror på vilket projekt det är och om kunden kräver det eller inte. De brukar utgå från känsla när de validerar och har inget ramverk som de följer.

4.7 Testa säkerheten

Cybercom har ett säkerhetsteam i Stockholm som ansvarar för att applikationerna publiceras med den tänkta säkerheten. Det finns även ett team i Malmö som testar applikationerna som Cybercom i Malmö utvecklar. Detta team testar vanligtvis om applikationen inte kan utsättas för vanliga typer av hot som de förväntar sig att applikationen inte ska kunna utsättas för, som exempelvis script injection. Både teamet i Malmö och det i Stockholm har ett protokoll som de följer när arbetar med applikationer.

När applikationer testas så testar de inte alla delar av applikationen lika mycket. De delar av applikationen som kräver mer säkerhet och har mer känslig datan testas mer än delar som inte har det.

I vissa projekt så testar kunder till Softhouse applikationerna i acceptansfasen. I sådana fall finns test inte med i kravspecifikationen och då utför Softhouse inte tester. Dock utförs alltid lågnivåtester, unit tester i alla projekt. För det mesta så sker säkerhetstester mot servern för att säkerställa att den kan ta emot och förhindra attacker. Detta på grund av att applikationen ofta är ett skal och inte innehåller värdefull information, men även för att användare inte kan utföra handlingar snabbt nog på en applikation för att få applikationen att fallera. De använder sig av verktyg för att testa säkerheten mot server men inte mot själva applikationen. Det är svårt för Softhouse att sätta upp testramverk för mindre projekt. Exempel på standarder som brukar testas vad gäller applikationerna är att man simulerar data, utför exploratory testing och ser till att flödet fungerar.

Anonym testar sina applikationer, men anser att de behöver göra det mer än vad de redan gör. De har två testare och fyra utvecklare. De anser att detta är för få testare och planerar att anställa fler testare. Anonyms utvecklare utför unit tester, vilket testarna ibland följer upp. Testarna är med i planeringsfasen i varje projekt, där de bygger upp sina testfall. De testar sen utifrån de testfall som har skapats i planeringsfasen. Det har hänt att Anonym har tagit hjälp av säkerhetsbolag som validerar de tjänster som Anonym har utvecklat, för att säkerställa att tjänsterna är säkra.

Xdin testar sina applikationer med automatiserade testfall och med happy-testing. Hur mycket de testar beror på hur mycket kunder efterfrågar test. Vanligtvis brukar de inte efterfråga test och i många fall testar kunderna själva applikationen. Dock har Xdin inte som vana att testa säkerheten i sina applikationer. Hur Xdin testar styrs mycket av hur man tidigare har gjort. Xdin har inget ramverk för att testa men de har vissa praxiser som har varit med i företaget under en tid som de följer. Utöver detta så använder de sig av vad de tycker är rimligt. De kan även följa kunders ramverk om kunderna har några och kräver att de används.

4.8 Säkra praxiser

Cybercom använder sig av SCRUM i alla projekt. Utöver detta så använder de sig inte av något ramverk eller metod när de utvecklar sina applikationer. Dock har både testteamet i Malmö och säkerhetsteamet i Stockholm protokoll som de följer.

Metodologier som Softhouse använder sig av är Test Driven Development och SCRUM. Test Driven Development försöker de använda i så stor grad de kan. De följer en egen tolkning av SCRUM och använder sig av den eller andra metoder som påminner om den. Enligt Softhouse så följer ingen SCRUM fullständigt, alla gör en egen version av SCRUM och följer den istället, vilket är varför de inte följer SCRUM fullständigt. De använder sig av dessa metoder för att få så korta leveranser som möjligt. Softhouse försöker även att använda sig av Continous Integration i alla sina projekt. I de mobila projekten kallas detta för Continous Deployment. En annan metodik som de försöker införa är en agil version av Microsofts Secure Development Lifecycle. Dock har de endast använt delar av den i tidigare projekt. På grund av att de mobila projekten ofta är kortlivade så anser Softhouse att de inte behöver använda sig av hela Secure Development Lifecycle, då det inte alltid är värt att föra in alla steg. I dessa fall implementerar de istället en vattenfallsmetod. Vilka metoder de använder sig av beror på från projekt till projekt. I vissa projekt använder de mindre metoder och i andra mer. Den största faktorn som bestämmer vilka metodologier som Softhouse använder sig av är projektets livslängd. Om det är ett större projekt så implementerar de vanligtvis ett större antal metodologier och om det är ett mindre projekt så implementerar de oftast ett mindre antal metodologier. Dock är SCRUM och Test Driven Development de två vanligaste metodologierna som Softhouse använder sig av.

Anonym använder sig av SCRUM i alla sina projekt. Dock har de en egen tolkning av metoden och använder sig inte av den fullständigt, vilket resulterar i att deras metod liknar SCRUM. De applicerar även en form av vattenfallsmetod på sina projekt, men de kör projekten som SCRUM-projekt. De återanvänder en del av den kod som de tidigare har skapat som de vet är säker. Anonym tror att deras agila arbetssätt gör att de är bättre på att prioritera och på sätt förbättras projektleveransen och säkerheten i varje projekt.

Anonym har även en modell för hur de ska implementera säkerhet i sina projekt. Detta är en stor modell som täcker väldigt mycket. De använder sig oftast inte av hela modellen utan endast delar av den som de anser vara viktiga för deras specifika projekt. Dock är denna modell utvecklad för en plattform, iOS, vilket Anonym inte är nöjda med. De anser att ett problem är att strukturen i säkerheten hos de olika hårdvaruutvecklarna gör det svårt för Anonym att skapa en liknande modell för Android.

Likt tester så har Xdin, i utvecklingsfasen, inga nedskrivna metoder, riktlinjer eller ramverk som de följer när de utvecklar sina applikationer. De följer de arbetssätt som tidigare använts. De anpassar sig efter vad kunden vill ha och om kunden kräver ett specifikt ramverk så följer Xdin detta, annars utvecklar de efter hur de känner är mest bekvämt.

4.9 Användning av verktyg

Cybercom använder sig av färdiga paket som verktyg under utvecklingen. Ett exempel på ett färdigt paket är Pro-Guard som de använder sig av när de utvecklar Android-applikationer. Pro-Guard hjälper de att se till att applikationen blir svår att avkoda och den skyddar källkoden. Ett annat exempel är att Apple har säkerhet inbyggt i processen att skapa applikationer för iOS. De har ett bugghanteringsprogram som heter JIRA där allt som görs i projektet kan följas upp och de har även GIT som arkiv.

Softhouse använder sig av verktyg när de testar servern, men inte när de testar applikationerna. Det enda verktyget de använder sig av när det gäller applikationer är ett verktyg som testar gränssnittet.

Anonym använder sig inte av några verktyg för att testa säkerheten i sina applikationer. De använder sig istället av säkerhetsbolag som validerar säkerheten i deras applikationer, när de anser att de behöver ta hjälp av säkerhetsbolag.

Xdin använder sig inte av några verktyg för att förbättra säkerheten, men om kunder kräver att ett används så använder de sig av det verktyget.

4.10 Säkerhetsmål

Cybercom definierar inte säkerhetsmålen i början av sina projekt. Säkerhetsmålen uppkommer under projektets gång. Under projektets gång upptäcker de var säkerheten måste vara och hur den ska utvecklas. Cybercom definierar inte mål i början av sina projekt eftersom de försöker lägga all känslig data i servern och undvika att applikationen innehåller känslig data.

Cybercom kartlägger inte hoten i sina applikationer eftersom de försöker hålla känslig data borta från applikationerna och lägga det på back-end istället. På grund av detta så anser de att datan i deras applikationer inte är känslig nog för att kartläggning eller modellering av hot krävs.

Säkerhetsmålen definieras ofta av kunder till Softhouse. Säkerhetsmålen finns med i kundernas kravställning och Softhouse brukar sällan öka kraven som ställs på dem. Anledningen är att det gör att projekten tar längre tid att utföra och en konsekvens att det är att det kostar mer för deras kunder. Kunden måste godkänna Softhouses arbete och om kunden inte är intresserad av säkerhet så går Softhouse vidare enligt kundens krav. Oftast brukar kundernas krav inte vara konkreta och en stor del av Softhouses projekt är att definiera exakt vad det är som kunden vill ha, för att sen kunna gå vidare med utvecklingen. Det händer även ofta att säkerhetsmål dyker upp senare i projekten. Hur säkerhetsmålen läggs upp beror på kunden och Softhouse försöker anpassa sig efter vad kunden vill ha och kundens tekniska förståelse.

I större projekt modellerar Softhouse hot och försöker kartlägga var och hur attacker kan ske och hur de kan skydda sig från dessa. De modellerar hot med hjälp av två olika modeller. Den fösta typen av modeller görs med hjälp av Data Flow Diagrams för att kartlägga dataflödet mellan olika dataentiteter i applikationen för att förstå var och hur datan är mottaglig för attacker. I de fall då Softhouse anser att det inte finns något mervärde i Data Flow Diagrams brukar de använda sig av sekvensdiagram. Sekvensdiagrammen använder de sig av för att få en översiktlig bild över applikationen medan Data Flow Diagram hjälper dem att få en inblick i hur applikationen fungerar på en detaljerad nivå.

Anonym definierar säkerhetsmål i början av sina projekt om projektet och kunden har säkerhet som ett krav. Detta görs tillsammans med kunden. Om det skulle vara så att varken projektet eller kunden kräver säkerhet så har Anonym en standard form av säkerhet som de följer. Det händer att kunden inte har konkreta säkerhetsmål, vilket gör att Anonym rådger kunder och på så sätt kommer fram till en kravspecifikation, som delvis innehåller säkerhet.

Anonym kartlägger inte hot i början av sina projekt. De har istället en modell som de vet är säker och täcker många hot och attacker. Denna modell använder de sig av när de anser att det behövs.

Xdin definierar säkerhetsmål i början av sina projekt tillsammans med resten av kravspecifikationen. Vilka målen är beror på vad kunden efterfrågar. De sätter upp en lista på säkerhetsmål som ska finnas i projektet. Dock brukar kunderna bortprioritera vissa säkerhetsmål på grund av att de tar lång tid och kostar mer att realisera.

Xdin kartlägger inte hot för sina applikationer.

4.11 Kunskap

Cybercoms utvecklare utbildar sig inte inom säkerhet. De anpassar sin kunskap utifrån varje projekt. Om de anser att deras kunskap brister så ser de till att de anskaffar sig den kunskap som behövs. De brukar ta hjälp av säkerhetsteamet i Stockholm, som hjälper dem att ta reda på vad varje projekt kräver.

När det gäller hot så håller sig Cybercoms anställda informerade och uppdaterade på egen hand. Deras intresse för teknik gör att de följer nyheter för sina respektive plattformar. De håller sig inte direkt uppdaterade om säkerheten, istället håller de sig uppdaterade om vad som sker allmänt och på så sätt blir de även informerade om säkerheten. Exempel på källor som de använder för att hålla sig uppdaterade IDG, Macrumours och The Loop. När det gäller verktygen de utvecklar i så håller de sig uppdaterade genom att alltid ha de senaste versionerna av verktygen. Dock kan vissa projekt kräva att de använder sig av äldre versioner av ett verktyg, vilket exempelvis kan vara på grund av att en kund kräver att man använder sig av en äldre version än den senaste. Cybercoms inställning till utvecklingsvertyg är att de bör hålla sig så uppdaterade som möjligt.

Softhouses säkerhetsexpert har haft kurser och utbildat utvecklare på Softhouse om säker mjukvaruutveckling. Dessa kurser har deras mobila team deltagit på. Kurserna behandlar olika aspekter av säkerhet och är till för att ge utvecklarna kunskap om hur man utvecklar säker mjukvara. Denna utbildning ska senare ligga som grund för de säkerhetsbeslut som utvecklare tar i sina olika projekt. Detta är den enda utbildningen Softhouse erbjuder sina utvecklare.

Softhouse har inga kontroller på att deras utvecklare håller sig uppdaterade vad gäller säkerhet. Det finns en överenskommelse mellan utvecklarna om att varje utvecklare ansvarar för att hålla sig uppdaterad inom den miljön de utvecklar för. Detta handlar delvis om säkerhet och de utvecklingsverktyg som de använder sig av. Det finns ett antal anställda på Softhouse som håller sig uppdaterade i denna fråga. De för vidare den information de anskaffar sig och på sätt ser de till att alla utvecklare tar del av ny och likadan information vad gäller säkerhet. På grund av att det finns många olika plattformar och miljöer så är det svårt för dessa få anställda att hålla alla uppdaterade om alla miljöer och plattformar, vilket gör att ansvaret till stor del läggs på utvecklarna.

Anonym erbjuder sina anställda utbildningar inom säkerhet om de anser att det behövs. Dock har de inte haft många utbildningar då deras konsulter är erfarna. De litar på att deras

konsulter har kunskap och håller sig uppdaterade om säkerhet. Dock utbildar de sina konsulter mycket när det gäller säkerhet i servern.

Anonyms utvecklare ansvarar själva för att hålla sig uppdaterade om vilka hot och attacker som existerar. De gör detta genom att läsa, exempelvis olika forum. Utvecklarna håller även sina utvecklingsverktyg uppdaterade.

Xdin erbjuder inte sina anställda utbildningar inom säkerhet. Företagets utvecklare håller sig uppdaterade om hot genom bloggar, forum och liknande. Varje utvecklare ansvarar för att hålla sig själv uppdaterad. När det gäller utvecklingsverktygen så håller sig Xdins utvecklare uppdaterade enligt vad de känner är rimligt. Dock kan vissa projekt även kräva att man använder sig av äldre versioner av utvecklingsverktyg.

4.12 Övrigt

Cybercoms kunder efterfrågar inte säkerhet. Cybercom får ofta försöka få kunderna att acceptera att säkerhet måste vara en del av projektet. Det beror ofta på vilken kund det är, de flesta efterfrågar inte säkerhet samtidigt som det finns vissa kunder, så som banker, som har säkerhet som ett stort krav på applikationen.

Softhouse ser ofta applikationer som skal där viktig data sällan lagras. En stor del av säkerheten brukar läggas på servern, vilket gör att säkerheten inte alltid får stor uppmärksamhet i applikationerna.

Anonym arbetar för det mesta med kunder som efterfrågar säkerhet till stor del. Detta gör att det ofta inte räcker med vanliga standarder som redan existerar. Anonym måste göra mer utöver de standarder som redan existerar som de använder sig av för att säkerställa att deras applikationer är säkra. Den färdiga modellen som Anonym har är en modell de själva skapat och säljer som ett stort färdigt paket. De använder sig inte alltid av hela modellen men de har den ifall den skulle behövas. Anonym väljer att kryptera allting som finns i deras system för att förhindra att någon kommer över känslig information.

Anonym har inställningen att allting inte behöver vara säkert, men att det alltid bör finnas en miniminivå som alla applikationer bör uppfylla.

Xdins kunder brukar efterfråga säkerhet i början av projekt, men prioriterar bort säkerheten mot slutet. Företaget anpassar sig efter vad kunden efterfrågar vilket gör att deras utvecklingsprocess inte innehåller mycket färdiga komponenter, metoder eller arbetssätt. De har inställningen att kunden alltid har rätt, men försöker få kunden att inte ta beslut som kan ha stora negativa konsekvenser vad gäller säkerheten.

4.13 Sammanfattning

För att sammanfatta resultatet av vår empiriska undersökning så har vi valt att skapa en tabell. Denna tabell är formad efter de tre tabeller som tillsammans utgör vårt teoretiska ramverk.

Som det tydligt går att se i tabellen nedan så använder sig alla fyra företag som vi intervjuat av det mesta som nämns i vår litteratur. Dock skiljer det sig väldigt mycket i hur företagen

inkluderar dessa faktorer i utveckling av sina mobilapplikationer. I nästa kapitel kommer vi att diskutera dessa skillnader.

Sammanfattning av empiri					
Faktorer		**Företag**			
Kategori	**Faktor**	**Cybercom**	**Softhouse**	**Anonym**	**Xdin**
Säkra åtgärder	Kryptering av kommunikation	X	X	X	X
	Autentisering	X	X	X	X
	Säker lagring av data	X	X	X	X
	Validera input	X	X	X	X
	Testa säkerheten	X	X	X	X
Säkra arbetssätt	Säkra praxiser	-	X	X	-
	Användning av verktyg	X	-	-	-
Säker kultur	Säkerhetsmål	X	X	X	X
	Kunskap	X	X	X	X
Summa:	**9**	8	8	8	7

Tabell 5: Frekvens av faktorer i den empiriska undersökningen.

5 Analys och diskussion

För att analysera vår empiriska undersöknig så har vi valt att jämföra våra resultat med vår litteratur. Vi kommer att jämföra de två för att se om företagen inkluderar de faktorer som litteraturen tar upp. Vi kommer även att jämföra de två för att ta reda på om företagen inkluderar faktorerna på samma sätt som litteraturen beskriver. Därefter kommer vi att inkludera egna tankar som baseras på vår litteratur och på de svar vi har fått från våra respondenter.

5.1 Kryptering av kommunikation

I vår litteratur märks det tydligt att kryptering av kommunikation är en viktig faktor när det gäller utveckling av säkra mobilapplikationer. La Polla, Martinelli & Sgandurra (2012) skriver om hur enkelt det är att använda sig av kryptering samtidigt som Hoog (2011) skriver om hur enkelt det är för angripare att komma åt data som inte är krypterad.

Alla fyra företag krypterar data som skickas till och från deras applikationer. HTTPS och SSL är de mest använda medlen för att kryptera kommunikation, vilket även är de mest omnämnda i vår litteratur. Anonym och Cybercom har som vana att kryptera kommunikation till och från applikationer medan Softhouse och Xdin först säkerställer att kryptering vid kommunikation krävs. Dock krypterar inte Cybercom all kommunikation då de försöker att inte ha känslig data i applikationen utan istället försöker placera den i back-end. De följer Kadrichs (2007) råd att försöka ha all viktig data borta från applikationen. Alla respondenter använder sig av redan existerande standardramverk, men Anonym är det enda företaget som inte litar på dessa ramverk och istället utvecklar egna metoder för kryptering. Anonym är det enda företaget, av de som vi har intervjuat, som följer Dwivedis (2010) råd. Han säger att mobilenheternas egna säkerhet är det första som angripare attackerar och under intervjun berättade Anonym att de inte enbart förlitar sig på de standarder som existerar på grund av att angripare oftast först attackerar dessa standarder. Om datan är känslig nog så har de alltid underliggande kryptering, vilket rekommenderas frekvent i vår litteratur. Samtliga respondenter anser att kryptering av kommunikation är viktig om datan är känslig. Dock är Anonym det enda företaget som har inställningen att det är bättre att kryptera mer än nödvändigt än mindre.

Vi har uppfattat det som så att företagen ser kryptering av data som skickas till och från mobilapplikationer som en viktig del utav utvecklingsprocessen. Vi har upptäckt att det är vanligt att kryptera kommunikation och vi tror att det beror på att de standarder som existerar är, vilket även nämns i vår litteratur, enkla att implementera vilket gör att många hot och attacker kan förhindras på ett enkelt sätt. Vi tror att företagen ser det som en fördel att enkelt kunna skydda sig mot många typer av hot utan att det tar mycket av projektets totala tid.

5.2 Autentisering

Även autentisering har visat sig vara en viktig faktor för säkerhet i vår litteratur. Dwivedi (2010) skriver att det är viktigt för applikationen att veta vem den kommunicerar med och att kunna säkerställa att de kommunicerar med rätt entitet. Om mobilapplikationen inte vet vem den kommunicerar med så kan en angripare enkelt komma åt känslig data.

Alla våra respondenter använder sig av autentisering, men sättet de gör det på varierar. Cybercom utför endast enkel autentisering, så som lösenordsvalidering, och litar på att back-end har säkerhetsmekanismer som kompenserar för detta. Detta strider mot Kadrichs (2007) åsikt om att servern borde skyddas från applikationen så mycket som möjligt genom att säkerhetsåtgärder implementeras i applikationen. På grund av att Cyberom låter servern

ansvara för största delen av säkerheten så lägger de inte mycket energi på att skydda servern från applikationen på bästa sätt. Softhouse kontrollerar noga att certifikaten som används i deras projekt är pålitliga. Detta stämmer överens med, bland annat, Hoog & Strzempka (2011) som skriver att det är viktigt att verifiera certifikaten. Anonym är det mest seriösa företaget vad gäller autentisering då de utvecklar egna processer för autentisering på grund av att de inte litar på de standarder som redan existerar. Detta stämmer, likt Softhouses autentisering, väl överens med det Hoog & Strzempka (2011) skriver. Det stämmer även väl överens med Gunasekera (2012) som skriver att företag bör utveckla sina egna certifikat för att vara säkra på att de är pålitliga. Xdin har stor tillit till sina utvecklare när det gäller autentisering då de låter utvecklarna själva bestämma hur de ska autentisera.

Likt kryptering av kommunikation så har vi i vår studie upptäckt att autentisering är vanligt vid utveckling av mobilapplikationer. Skillnaderna ligger i hur autentiseringen implementeras. Vissa företag har en mer seriös inställning till autentisering samtidigt som andra företag inser att det bör finnas med, men de spenderar oftast inte mycket energi på att säkerställa att det sker på rätt sätt. Företagen med en mindre seriös inställning till autentisering litar på de standarder som existerar, de kontrollerar inte att autentiseringen är säker och de säkerställer inte att autentisering implementeras, utan litar på att utvecklarna gör detta. I enlighet med vår litteratur så kan befintliga standarder räcka till, men de är inte det säkraste alternativet.

5.3 Säker lagring av data

I vår litteratur har vi funnit att data även är sårbar när den inte transporteras. Dwivedi (2010) skriver att data som lagras är sårbar om den inte lagras på ett bra sätt. Att lagra data på ett säkert sätt är, enligt honom, viktigt om man vill skydda sin känsliga data från angripare. Zdziarski (2012) skriver att all data som lagras lokalt, och inte i servern, måste krypteras.

Resultatet av vår studie visar att det är vanligt att mobilapplikationsutvecklare lagrar data enligt den kontext som redan existerar på plattformen som de utvecklar för. Cybercom, Softhouse och Xdin använder sig av plattformens egna säkra lagring. Miller et al. (2012) anser att detta inte är bra om datan är känslig eftersom de inbyggda lagringssystemen oftast inte räcker till. Anonym använder sig inte av enhetens egna lagringssystem, utan utvecklar själv egna sätt att lagra datan säkert, och deras motivering till detta beslut är likt det Zdziarski (2012) skriver om att angripare oftast attackerar den befintliga lagringen först. Zdziarski (2012) skriver också att alla utvecklare själva bör kryptera datan ifall den verkligen är känslig vilket alla respondenter, utöver Anonym, strider mot. Endast Anonym har inställningen att det är bättre att kryptera mer data än mindre. De andra tre respondenterna krypterar endast data som de anser behöver krypteras. Alla respondenter nämner att de alltid har i åtanke vad det är för applikation som utvecklas, vilket överensstämmer med Kadrichs (2007) åsikter om att det är viktigt att ta hänsyn till om säkerhet behövs eller inte.

Alla respondenter lagrar data säkert till viss grad. Vi tror att beslut om hur data ska lagras, utöver kundernas krav, till stor del formas av de befintliga verktygen som existerar för säker lagring av data. Detta eftersom tre av fyra respondenter litar på dessa befintliga metoder och utvecklar sällan egna. Tre av fyra respondenter tycker att applikationen endast fungerar som ett skal och har inställningen att känslig data inte existerar i applikationen. När känslig data existerar i applikationen så implementerar de en snabb standard som är enkel att använda, endast på grund av att de inser att känslig data inte bör ligga okrypterad på den mobila

enheten. Ett av fyra företag inser vikten av att anstränga sig för att skydda känslig data i applikationen och utvecklar egen säker lagring. Vår litteratur nämner att det börjar bli mer vanligt för applikationer att hantera känslig data, vilket Rassan & Al Sheik (2013) exemplifierar med att nämna information om bankkonton. La Polla, Martinelli & Sgandurra (2012) skriver om att mer och mer data har börjat lagras lokalt i applikationer. Av våra respondenter är endast en redo för en ökad efterfrågan på säkerhet, medan de andra måste ändra på sina rutiner för att anpassa sig.

5.4 Validera input

Validering av input är en av de punkter som, enligt vår litteratur, är extremt viktigt för säkerheten i mobilapplikationer. Six (2011) skriver att utvecklare bör anta att all input är farlig, vilket innebär att all input bör valideras. Detta gör att det är enkelt att förhindra attacker som, för angriparen, är enkla att utföra. Han skriver även att validering av input är grunden till säkerhet i mobilapplikationer.

Alla företag i vår studie validerar input, men sättet de validerar input på skiljer sig mellan företagen. Cybercoms utvecklare utför endast enkel validering av input i applikationen eftersom de försöker placera all säkerhet i servern. Detta strider mot Kadrichs (2007) åsikter om att applikationen bör ha mycket säkerhet i sig för att skydda servern. Softhouse validerar all input genom att de erbjuder utbildningar till sina utvecklare. Dessa utbildningar ska ligga till grund för de beslut som tas av utvecklarna och en av dessa beslut behandlar validering av input. Softhouse litar på att deras utvecklare validerar input korrekt. Anonym validerar all input och Xdin litar, likt Softhouse, på att deras utvecklare validerar input korrekt. Dock erbjuds Xdins utvecklare inte utbildning där validering av input ingår. Alla företag validerar input i applikationen men Cybercom har större tillit till serverns säkerhet jämfört med de andra företagen. I vår litteratur diskuteras det hur man kan validera input och vilket sätt som är det mest säkra. Ingen av de företag som vi har intervjuat använder sig av ett specifikt sätt att validera input på utan de låter istället utvecklarna själva bestämma hur input valideras vilket innebär att det inte finns en standard för hur input valideras. Reject-known-bad och accept-known-good är två exempel som tas upp av Six (2011).

Vi har upptäckt att validering av input är vanligt hos våra respondenter och vi tror att tre omständigheter styr hur mycket företag validerar input. För det första så kan en input validation attack ske enkelt, vilket inte kräver stor ansträngning från angriparens sida (Basavala, Kumar & Agarrwal, 2013). För det andra så har denna typ av attacker visat kunna skapa stor skada (Son, Lee & Oh, 2012). Sist är själva enkelheten i att validera input (Kadrich (2007). Dock litar företagen väldigt mycket på sina utvecklare och deras kompetens.

5.5 Testa säkerheten

Dunham (2008) skriver att det är viktigt att säkerheten i mobilapplikationer testas. Hoog (2011) påpekar att om man ämnar skapa en säker mobilapplikation så är test av säkerheten en nödvändighet. Kadrich (2007) skriver att test är den sista chansen att upptäcka fel innan någon annan med skadliga avsikter gör det och utnyttjar det.

Som förväntat, utifrån hur mycket det nämns i litteraturen, så testar alla företag i vår studie sina applikationer. Cybercom har ett testteam i Malmö och ett säkerhetsteam i Stockholm vilket gör att Cybercom tar säkerhet på stort allvar. Detta stämmer överens med Kadrichs

(2007) rekommendation om att den som utvecklar inte bör vara den som testar. Softhouse testar endast när kunden kräver det, ofta testar också kunden själv. De har dock minimikrav som de själva alltid testar i alla projekt. Hoog & Strzempka (2011) anser att ett minimumkrav för hur mycket som testas är viktigt att definiera. Anonym testar sina applikationer med hjälp av de testare som finns anställda på företaget, vilket de tyckte var för få, och med hjälp av säkerhetsexperter som underleverantörer, vilket likt Cybercom stämmer väl överens med vad Kadrich (2007) skriver. Det stämmer även överens med det Basavala, Kumar & Agarrwal (2013) påpekar när de rekommenderar att man använder sig av andra företag för att testa säkerheten. Anonyms testare är med i början av projekten när säkerhetskraven ställs och utför sina tester efter de testfall som då skapas. Fried (2010) påpekar att det är viktigt att alla i ett utvecklingsprojekt har samma bild av säkerheten. Att testare är med när säkerhetskraven ställs gör att även de är medvetna om säkerheten i projektet. Anonym tar inte ofta hjälp av sina underleverantörer när det gäller säkerhet medan Cybercom har ett säkerhetsteam i Stockholm som kan hjälpa de väldigt mycket och ofta. Xdin testar, likt Softhouse, efter vad kunder kräver och eftersom kunder sällan kräver säkerhet så testar företaget inte ofta säkerheten i sina applikationer. När säkerhet är ett krav så testar de den.

Eftersom test är en sista kontroll innan applikationen levereras till kunden (Kadrich, 2007) och med tanke på hur ofta det nämns i vår litteratur så är det inte förvånande att alla respondenter testar sina applikationer. Dock är det förvånande att två av våra respondenter inte testar applikationers säkerhet noggrant på grund av att deras kunder inte efterfrågar säkerhet. Vi tror att kunder, utifrån de svar vi har fått, inte efterfrågar säkerhet på grund av att de inte är medvetna om vilka risker som existerar och vilken skada en attack som säkerheten i applikationen kan medföra. Säkra applikationer utvecklas inte om kunder inte är intresserade av säkra applikationer. Våra respondenter ger kunder råd gällande säkerhet men i slutändan så är det kunden som bestämmer, vilket vi inte finner överraskande eftersom våra respondenter är konsultföretag. Dock finner vi det överraskande att kunder inte efterfrågar säkerhet, speciellt med tanke på de konsekvenser som dålig säkerhet kan leda till som nämns i vår litteratur.

5.6 Säkra praxiser

Vår litteratur visar att användning av säkra praxiser underlättar arbetet med säkerhet till stor del. Säkra praxiser är uppbyggda av mycket erfarenhet, kunskap och forskning vilket innebär att de bidrar med mycket. Zdziarski (2012) anser att säkra praxiser kan hjälpa till med att se till att säkerhetsproblem aldrig uppstår. De kan vara till stor hjälp för utvecklaren och Hoog & Strzempka (2011) anser att utvecklare, inom en snar framtida, kommer vara tvungna att använda sig av säkra praxiser.

Vi har upptäckt att SCRUM är vanligt hos våra respondenter i och med att Cybercom, Softhouse och Anonym använder sig av det. Enligt Six (2011) är detta positivt eftersom agila metoder har visat sig vara de mest gynnsamma för säkerheten. Det enda företaget som inte använder sig av SCRUM är Xdin, som för övrigt inte använder sig av några metoder överhuvudtaget, om kunden specifikt inte ber om det. Xdin har istället valt att varje utvecklare själv får bestämma hur denne ska utveckla, vilket oftast är format efter hur man tidigare har utvecklat. Hoog (2011) anser att utvecklare kan använda sig av tidigare arbetssätt som har fungerat bra, så länge de är resulterar i säker mjukvara. Kadrich (2007) säger att en säker standard är av största vikt vid utveckling av applikationer som är tänkta att vara säkra. Det är möjligt att Xdins utvecklare följer säkra praxiser men företaget kan inte garantera att alla

deras applikationer utvecklas efter säkra praxiser. Anonym använder sig inte av några säkra praxiser men de har mycket färdig kod som tidigare har använts och som de garanterar är säker. Denna kod brukar de återanvända för att spara tid och garantera säkerhet, vilket överensstämmer med det Hoog (2011) skriver om att återanvända tidigare säkra arbetssätt. Anonym återanvänder inte ett arbetssätt utan de återanvänder istället ett arbete, vilket enligt oss är detsamma som att återanvända ett arbetssätt. Alla som använder sig av SCRUM använder sig av en egen tolkning av SCRUM. Cybercoms testare har även färdiga protokoll som de följer när de testar. Softhouse använder sig, utöver SCRUM, av Test Driven Development, Continous Integration och Microsofts Secure Development Lifecycle beroende på vad det är för projekt och vad kunden efterfrågar. På grund av att Softhouse är det enda företaget som, förutom att använda sig av en agil metod, även använder sig av en metod som är utvecklad specifikt för att skapa säker mjukvara så överträffar de resten av våra respondenter när det gäller användning av säkra metoder. Anonym använder sig, utöver SCRUM, av en egen vattenfallsmodell.

Säkra praxiser är, bland våra respondenter, vanliga. Även om tre respondenter använder sig av SCRUM så har de även andra ramverk eller metoder som de följer som hjälper de med säkerheten. Dock är det endast två företag som kan garantera att alla deras applikationer utvecklas med hjälp av säkra arbetssätt. Vi tror att företagen väljer att ha metoder för att de, vilket även nämns i vår litteratur, är enkla att använda och är till stor hjälp. Säkra praxiser hjälper utvecklare att inte missa viktiga punkter och att hantera fel när de väl uppstår (La Polla, Martinelli & Sgandurra, 2012). Vi tror att, i de fall då säkra praxiser inte används, så har kunderna stor påverkan. En kund som inte är intresserad av säkerhet vill inte att företag arbetar efter säkra praxiser. Svaren från våra respondenter påvisar att kunder är ovilliga att betala mer än nödvändigt.

5.7 Användning av verktyg

I vår litteratur tas många exempel upp över hur olika verktyg kan användas för att hjälpa utvecklare att skapa säkra mobilapplikationer. Dessa verktyg kan hjälpa utvecklaren både med att utveckla säkerhet men även att säkerställa att den utvecklade säkerheten är säker. Zdziarski (2012) påpekar att verktyg kan förenkla och påskynda processen av att skapa säkra mobilapplikationer, vilket är gynnsamt för konsulter och deras kunder ur en tid- och kostnadsperspektiv.

Cybercom använder sig av färdiga paket när de utvecklar sina applikationer. Två exempel är Pro-Guard och JIRA. Enligt Six (2011) så bör verktyg användas. Ett av hans exempel handlar om verktyg som obfuskerar källkoden, vilket Pro-Guard kan användas till att göra. Softhouse använder sig av verktyg när de arbetar med servern men inte när de arbetar med applikationer, vilket likt mycket annat som Cybercom gör strider mot Kadrichs (2007) åsikter om att man bör skydda servern från applikationen så mycket som möjligt. Anonym använder inga verktyg för att förbättra säkerheten, de använder sig istället av säkerhetsbolag för att säkerställa att applikationerna är säkra, vilket som tidigare nämnts rekommenderas i vår litteratur. Xdin använder sig inte av verktyg om kunden inte specifikt kräver att verktyg används.

Vår litteratur visar, med hjälp av många exempel, hur verktyg kan påskynda och förenkla utvecklingen av säkerhet och det rekommenderas frekvent. I vår studie framkommer det att företagen vanligtvis inte använder sig av verktyg, förutom Cybercom som använder sig av ett fåtal verktyg. Utifrån tidigare svar då företagen har förklarat att de litar mycket på sina

utvecklare så tror vi att anledningen till att verktyg inte används är för att de inte ser ett mervärde av att använda sig av verktyg. De litar på sina utvecklares egen erfarenhet och kunskaper. Som tidigare nämnts så tror vi att ännu en anledning är att kunder inte är intresserade av att deras konsulter använder verktyg för säkerhet eftersom kunderna inte är intresserade av säkerhet och gärna inte vill betala mer än nödvändigt.

5.8 Kunskap

För att utvecklare ska kunna utveckla säkra applikationer så är det av största vikt att de är kunniga inom säkerhet (Kadrich, 2007). Utvecklare som inte är kunniga inom säkerhet kommer att göra fel när de utvecklar mobilapplikationer (Zdziarski, 2012). I vår litteratur så rekommenderas utbildning som ett bra sätt att anskaffa sig kunskap. Dwivedi (2012) skriver att utvecklare bör vara medvetna om vilka hot som existerar och hur man bör skydda sig från dessa.

Utbildning inom säkerhet är varken vanligt eller sällsynt hos våra respondenter. Softhouse och Anonym utbildar sina anställda medan Cybercom gör det sällan och Xdin inte gör det överhuvudtaget. Xdin litar, mer än de andra företagen, på sina utvecklare när det gäller säkerheten i deras applikationer och är ett av två företag som inte erbjuder någon utbildning inom säkerhet. Softhouse har en anställd säkerhetsexpert som ger kurser inom säkerhet för att säkerställa att utvecklarnas kunskap om hot och skydd är tillräcklig för att företaget ska kunna lita på deras omdöme vad gäller säkerhet i framtida projekt. Anonym utbildar sina utvecklare när de känner att det behövs. De lägger stor tillit på sina utvecklare och på säkerhetsbolaget som de anlitar till vissa projekt. Cybercom ser till att utvecklarna har den kunskap som krävs inför varje projekt innan de påbörjar projektet och oftast behövs ingen större utbildning för detta. Cybercom och Xdin litar på att deras utvecklare själva håller sig informerade och uppdaterade om vilka hot och risker som existerar och hur man skyddar sig från dessa. Dock har Cybercom ett säkerhetsteam i Stockholm som de kontaktar om de skulle behöva vägledning i ett projekt.

Samtliga företag litar på att utvecklarna själva håller sig uppdaterade om vilka nya hot som uppkommer och vilka nya sätt att förhindra dessa hot som existerar. Endast ett av företagen har någon form av utbildning inom nya hot och risker. Två av företagen utbildar sina anställda när de anser att det finns ett behov av utbildning. Softhouse har ett antal anställda som har tagit på sig ansvaret att hålla sig uppdaterade om vilka hot och risker som finns och förmedla detta vidare till alla utvecklare. Dock är det svårt för dessa få anställda att hålla sig uppdaterade inom alla miljöer och plattformar. I samtliga företag håller utvecklarna sig uppdaterade genom att läsa olika bloggar eller forum.

Utbildning inom säkerhet, enligt vår litteratur, är viktigt då en stor del av säkerhetsansvaret placeras på utvecklare. Om dessa inte är kunniga inom säkerhet så kan detta resultera i att applikationerna inte uppnår den graden av säkerhet som man tror att de gör. Ett exempel kan vara att företag är säkra på att deras applikationer inte är sårbara mot input validation attacker för att de litar på att utvecklaren har skapat säker kod som validerar all input. Om utvecklare inte förstår varför input bör valideras så kan det vara så att valideringen av input inte blir säker. Som tidigare nämnt så litar våra respondenter mycket på sina utvecklare. Även användning av verktyg är sällsynt. Trots att så mycket tillit placeras på utvecklarna så erbjuder endast ett av fyra företag utbildning inom säkerhet utöver de tillfällen då det anses vara nödvändigt. Vi tror att en anledning till att utbildning inom säkerhet inte är populärt

bland våra respondenter är för att utbildning kostar mycket och tar tid. Att utvecklare spenderar tid på att delta i dyra kurser inom ett ämne som kunder inte är intresserade av gör det förståeligt varför företag inte är mer intresserade av utbildning inom säkerhet.

Vi tror att företag, baserat på våra respondenters svar, inte anser att säkerhetshot utvecklas i en snabb takt. De känner att den utveckling som sker snabbt upptäcks av många och på så sätt delas informationen av nya upptäckter snabbt på nätet. Denna snabba spridning av information över nätet gör att företag litar på att bloggar och forum innehåller den information som utvecklare behöver för att hålla sig uppdaterade vad gäller säkerhet. Eftersom alla företag till stor del litar på att utvecklarna så blir det viktigare att utbilda utvecklare inom säkerhet för att öka deras förståelse för säkerhet. Hur väl utvecklarna själva håller sig uppdaterade kan vi inte spekulera om men Son Lee & Oh (2012) skriver att utvecklare med lite kunskap kan komplettera detta med att använda sig av verktyg, vilket inte är vanligt hos våra respondenter. Både företagen och vår litteratur anser att kunskap inom säkerhet är viktigt, dock litar företagen på att deras utvecklare redan har tillräcklig kunskap.

5.9 Säkerhetsmål

Säkerhet, i grunden, handlar om att veta vilka hot som finns och hur man skyddar sig från dessa. Detta kan göras genom att definiera säkerhetsmålen för ett projekt (Fried, 2010) eller modellera hoten för ett projekt (Basavala, Kumar & Agarrwal, 2013).

Definition av säkerhetsmål är någonting som oftast sker, enligt vår studie. Cybercom definierar inte säkerhetsmål i början av sina projekt. De litar på att säkerhetsmålen själva uppstår under utvecklingen genom att man under utvecklingens gång, som utvecklare, själv upptäcker var säkerhet behövs. Detta innebär att Cybercom tar sig an projekt utan att vara medvetna om vilka risker som finns och hur de ska skydda sig mot detta, vilket strider mot det som står i vår litteratur. Softhouse, Anonym och Xdin definierar säkerhetsmål i början av sina projekt tillsammans med kund och samtidigt som resten av kravspecifikationen skapas. Softhouse brukar lägga till säkerhetsmål om de anser att det behövs. Dock innebär detta inte att Softhouse inte är medvetna om vilka risker som finns. Softhouse anser att det är viktigt för dem att veta vilka risker som finns, även om de inte motverkar alla på begäran av kunden. Anonym har en standardram för hur säker varje applikation bör vara vilket är någonting som rekommenderas i vår litteratur. Xdin brukar dock ta bort vissa säkerhetsmål på begäran av kunder, vilket visar hur ointresserade kunder är av säkerhet.

Vår litteratur har visat hur mycket modellering kan hjälpa utvecklare att finna risker, men även på ett bättre sätt förstå säkerheten i applikationen. Modellering av hot är inte någonting som vanligtvis sker, hos våra respondenter, vid utveckling av mobilapplikationer. Cybercom, Anonym och Xdin kartlägger inte hot för sina projekt. Cybercom motiverar detta med att de försöker hålla känslig data borta från applikationen och placerar den istället i back-end. Om ingen känslig data existerar i applikationen så kan man inte modellera säkerheten eftersom den inte behövs. Dock påpekar Kadrich (2007) att säkerhet behövs i applikationer även om känslig data inte finns i applikationen. Servern kan bli attackerad via applikationen och extra säkerhet utöver den som finns i servern kan vara till stor hjälp för att skydda servern. Anonym kartlägger inte sina hot eftersom de har mycket färdig kod som de återanvänder och anser vara säker. De anser att användningen av detta ramverk, eller delar av det, gör att modellering av hot inte är nödvändig. Xdin har ingen motivering till varför de inte kartlägger hot. Softhouse är det enda företaget som kartlägger sina hot. De gör detta med hjälp av två olika

notationer. Vilken av dessa som används beror på hur stora säkerhetskraven är på projektet de arbetar med.

Resultatet av vår undersökning visar att säkerhetsmål oftast sätts upp tillsammans med resten av kravspecifikationen. Säkerhetsplanering isoleras inte utan sker tillsammans med resten av planeringen, vilket innebär att de följer rekommendationerna i vår litteratur. Säkerhet efterfrågas oftast inte av kunder, vilket också är en bidragande orsak till varför säkerhet inte planeras mer utförligt och varför säkerhetsplaneringen inte får mer uppmärksamhet.

Vi tror att anledningen till att företag inte använder sig av modellering av hot är för att de litar mycket på sina utvecklare. Detta tror på grund av att de ofta påpekar att du litar på att utvecklarna gör ett bra jobb när det gäller säkerhet. De ser inget värde i att modellera hot precis som de inte ser ett värde av att utbilda eller använda sig av verktyg. Ytterligare en anledning, tror vi, är att kunder inte vill betala för modellering, de vill endast betala för funktionalitet. De är intresserade av att betala så lite som möjligt och eftersom modellering tar tid och de inte ser er mervärde i att modellera hot så är de inte intresserade av kartläggning av hot.

5.10 Övrigt

Cybercoms, Softhouses och Xdins kunder efterfrågar vanligtvis inte säkerhet. Anonyms kunder efterfrågar säkerhet men deras förståelse för säkerhet begränsar deras säkerhetskrav. Samtliga företag säger att deras kunder inte har stor förståelse för säkerhet vilket gör att deras kravspecifikationer inte har med konkreta säkerhetskrav. Det är ofta upp till våra respondenter att föreslå säkerhetslösningar genom att försöka förstå vad kunderna behöver. Samtliga företag ser applikationen som ett mellanting mellan användare och back-end. Detta gör att deras bild av mobilapplikationer är att dessa endast fungerar som skal. Så lite känslig information som möjligt ska lagras i applikationen, vilket medför att det stora säkerhetsansvaret placeras på back-end. Ännu en gemensam nämnare för samtliga företag är att de anpassar sig mycket efter vad kunden kräver. De tvingar sällan kunden att ändra sina krav, de fungerar mer som rådgivare och kunden har alltid sista ordet.

6 Slutsatser

I detta kapitel kommer vi att presentera de slutsatser som vi har kommit fram till i vår undersökning. Slutsatserna kommer att baseras på de empiriska data som vi anskaffat oss, vilket innebär att vi inte kommer att dra några generella slutsatser gällande samtliga företag i vår region utan endast de fyra företag som vi har intervjuat. Detta kapitel ska svara på vår forskningsfråga samt sammanfatta vår undersökning.

Syftet med denna studie var att ta reda på, på vilket sätt företag inför säkerhet vid utvecklingen av mobilapplikationer, genom att undersöka vad de gör under utvecklingen av dessa applikationer och hur de gör det. Vi har identifierat nio viktiga faktorer i vår litteratur som bör finnas med i utveckling av säkra mobilapplikationer. Utifrån dessa faktorer har vi skapat ett teoretiskt ramverk indelat i tre kategorier som vi själva har skapat. Vi har använt oss av detta ramverk för att utforma intervjufrågor som vi sen har använt under intervjuer med fyra IT-konsultföretag i Malmö och Lund. Dessa frågor har lett till svar som beskriver hur dessa företag arbetar med säkerhet vid utvecklingen av deras mobilapplikationer. Vi har följt strukturen av vårt teoretiska ramverk när vi har presenterat och analyserat vår insamlade data från vår empiriska undersökning. Vi har jämfört företagens svar på våra frågor med vad som står i vår litteratur. Vi kan inte dra generella slutsatser gällande alla IT-konsultföretag i vår region eftersom vår inblick i dessa är begränsad till endast fyra företag. Dock anser vi att vår studie går att upprepas eftersom vår metod är tydligt strukturerad och enkel att följa.

När det gäller säkerhet så inkluderar alla företag majoriteten av de viktiga faktorer som tas upp i vår litteratur. Skillnaderna ligger i hur dessa faktorer inkluderas. Samtliga respondenters arbete med säkerhet är i de flesta punkterna, enligt vår litteratur, godtagbar. Vissa företag har en seriös inställning till säkerhet vilket leder till att de försöker göra mer än endast det som är godtagbart när det gäller säkerhet. Andra företag har endast med säkerhet för att de vet att det bör finnas med men försöker inte skapa säkerhet som är mer än godtagbar. Dock överensstämmer dessa företags bild av vad som är godtagbar säkerhet med vad som står i vår litteratur. Följaktligen så utvecklar våra respondenter, för det mesta, godtagbart säkra mobilapplikationer.

Hur företagen väljer att inkludera dessa faktorer påverkas av två externa orsaker och en intern. Den första externa orsaken är applikationen själv. Mer specifikt så är det volymen av känslig data och hur känslig datan är som styr om säkerhet behövs eller inte. Mer känslig data och högre grad av känslighet ökar kraven på säkerhet. Detta är ett förväntat resultat som nämns i vår litteratur. Den andra externa orsaken är kunderna och dess åsikter gällande säkerhet. Våra respondenters kunder är sällan intresserade av säkerhet och på grund av detta så begränsas utvecklingen av säkerhet i mobilapplikationerna. Den andra orsaken finns inte med i vår litteratur, men det var samtidigt inte ett resultat som förvånade oss eftersom våra respondenter är konsultbolag. Dock var det inte förväntat att kunderna skulle vara så motvilliga till att säkerhet tar plats under utvecklingen av applikationerna. Även om dessa två externa orsaker påverkar företagen så påverkar de endast hur företagen utvecklar säkerhet, dock eliminerar de oftast inte säkerheten helt från mobilapplikationen, eftersom företagen är medvetna om de risker som kan uppkomma ifall säkerhet inte existerar. Om känslig data hanteras av applikationen så inkluderas många av de nio viktiga faktorerna instinktivt av företagen. Företagen påverkas väldigt mycket av kundernas önskemål, vilket innebär att de oftast inte arbetar mycket med säkerhet på grund av att kunderna inte efterfrågar det men de är beredda att möta stränga och krävande önskemål gällande säkerhet, vilket tydligt har framkommit under intervjuerna.

Den interna orsaken som vi identifierade och som överlag påverkar företagens arbete med säkerhet är utvecklarna själva. Samtliga företag litar väldigt mycket på att deras utvecklare är kunniga inom säkerhet, tar goda beslut gällande säkerhet och implementerar säkerhet på ett bra sätt. I ett av företagen så är det utvecklarnas ansvar att ta reda på om säkerhet behövs genom att försöka finna möjliga risker under tiden som applikationen utvecklas.

Vår slutsats är att företagen har kapacitet och viljan till att utveckla säkra mobilapplikationer och oftast införs godtagbar säkerhet vid utvecklingen av dessa. Företagen är medvetna om att säkerhet behövs och försöker att utveckla så säkert som situationen tillåter och kräver. Det som, mer än annat, förhindrar applikationerna från att vara säkra är kunderna. Detta innebär att om säkerheten skulle brista så är det inte på grund av att utvecklare inte har kunskap eller kompetens till att utveckla säkra applikationer och det är inte heller på grund av att utvecklare är motvilliga till att skapa säkra applikationer. Anledningen är att kunder inte är intresserade av säkerhet och på sätt begränsas utvecklingen av säkerhet i mobilapplikationer.

Företagen försöker prioritera säkerhet, vilket tydligt visas i tabell 5, och ofta är säkerheten någonting som företagen själva försöker driva framåt i projekt. De säkerhetskrav som ställs kan företagen möta och de krav som inte ställs kan företag kompensera för. Det som gör att säkerheten brister och prioriteras bort är de krav som begränsar och förhindrar införandet av säkerhet i utvecklingen av mobilapplikationer.

7 Appendix

7.1 Bilaga 1 – Definitioner

Autentisering
En process där en entitet verifierar att en annan entitet är det den utger sig för att vara (Howard & LeBlanc, 2002).

Back-end
Bakomliggande mjukvara som stödjer och möjliggör handlingar som användaren utför i ett program (Howard & LeBlanc, 2002).

Buffer overflow
En form av attack där mer data sparas i det temporära minnet än vad det får plats med. Det resulterar i att angriparen kan exekvera skadlig kod när det temporära minnet försöker åtgärda bristen av minne (Howard & LeBlanc, 2002).

Brute force attacker
En typ av attack där angripare försöker gissa sig fram till en säkerhetsnyckel med vilken tillgång till känslig data möjliggörs (Howard & LeBlanc, 2002).

Dekryptera
En process där krypterad data omvandlas till dess ursprungliga format (Howard & LeBlanc, 2002).

Device authentication
Autentisering av en enhet (Howard & LeBlanc, 2002).

Encoding
Att omvandla data till ett annat format (Howard & LeBlanc, 2002).

Hash
En form av kryptering där ett meddelande omvandlas, med hjälp av aritmetik, till en lång siffra (Howard & LeBlanc, 2002).

HTTPS
Ett protokoll för att skicka och visa data på ett säkert sätt (Howard & LeBlanc, 2002).

Input
Input är data som skickas in i ett program från en utomstående entitet (Howard & LeBlanc, 2002).

Input validation attacker
En form av attacker där skadlig data skickas till och tas emot av ett program (Howard & LeBlanc, 2002).

Klient
En klient är en entitet som använder sig av en servers tjänster (Howard & LeBlanc, 2002).

Kryptering
Kryptering är en process där data omvandlas på så sätt så att den blir oläslig. Samma data kan omvandlas tillbaka till den ursprungliga formen, om angriparen har en nyckel som förklarar hur denne ska göra detta (Howard & LeBlanc, 2002).

Känslig data
Känslig data är data vars manipulation kan orsaka skada mot personer eller organisationer (Howard & LeBlanc, 2002).

Mannen-i-mitten attacker
En attack där en angripare placerar sig mellan två entiteter. När detta sker så sker all kommunikation mellan dessa två entiteter via angriparen. För entiteterna verkar det som att de kommunicerar direkt med varandra. Denna typ av attack gör det möjligt för angripare att fånga upp och manipulera känslig data i kommunikationen (Howard & LeBlanc, 2002).

Mobil malware
Ett program som är skapat för att utgöra digital skada (Howard & LeBlanc, 2002).

Molnet
En annan term för internet som ofta används för att beskriva tjänster som sker över internet (Dwivedi, 2010).

Reverse engineering
En metod för att återvinna källkoden av ett program med hjälp av maskinkod (Howard & LeBlanc, 2002).

Server
En server är ett mjukvarusystem som erbjuder tjänster till en eller flera klienter (Howard & LeBlanc, 2002).

Signera kod
En metod för att placera digitala signaturer i mjukvara (Howard & LeBlanc, 2002).

SSL
En standard teknik för att skapa säker krypterad kommunikation (Howard & LeBlanc, 2002).

SSL session
En serie av transaktioner av data mellan entiteter som sker med hjälp av SSL (Howard & LeBlanc, 2002).

SQL-injection
En form av input validation attack där indatan är i form av en databasfråga (Howard & LeBlanc, 2002).

TLS
En standard teknik för att skapa säker krypterad kommunikation (Howard & LeBlanc, 2002).

Tvåfaktorsautentisering
Autentisering som kräver två metoder för autentisering vid samma tillfälle (Howard & LeBlanc, 2002).

Unit tester
Automatiska tester av kod som utförs av ett program (Howard & LeBlanc, 2002).

User authentication
Autentisering av en användare (Howard & LeBlanc, 2002).

Zitmo - internet
En form av malware som är skapad för att bryta sig igenom tvåfaktorsautentisering (Dunham, 2008).

7.2 Bilaga 2 - Transkribering av intervju med Cybercom

Transkribering av intervju med Johan Enell, IT Developer på Cybercom

Berätta lite om dig själv, din bakgrund

Jag pluggade datalogi på Lunds Universitet, tog min examen 2004 därifrån, efter det började jag jobba på universitetet på Medicinska fakulteten och hjälpte dom med analys och beräkningar av deras forskningsmaterial där. Två år var jag där sen fick jag möjligheten att börja på Cybercom började som Java back-end utvecklare och även lite front-end GUI men låg på mina chefer att jag ville göra iPhone appar, så efter två år så fick jag möjlighet att börja med mobilapplikationsprojekt, två veckor hade jag på mig att lära mig iPhone sen körde vi igång projektet. Så sen dess har jag jobbat med iPhone utveckling enbart, faktiskt så hade jag lite Android appar i det senaste projektet men det har varit uteslutet iPhone appar jag jobbat med.

Hur Cybercom tar hänsyn till säkerheten när de utvecklar mobilapplikationer

Det är från fall till fall, att utveckla säkerheten är ju alltid en tidsfråga, hur mycket tid man har att lägga på det, men utifrån vad det är för applikation och hur känslig datan är så försöker vi ju anpassa säkerheten efter det. Och jag har väl lite grann inställningen att det finns ingenting man kan göra som är hundra procent säkert utan det gäller bara att göra det tillräckligt svårt så att man inte ska vara värt att lägga ner tiden, så a precis utifrån vad det är för data så bestämmer vi vad vi ska ha för säkerhetslösning och det kan vara allt ifrån att vi gör en liten hash-kryptering liksom en enkel handskakning mellan mobil och server, app och server till att vi tar in vårat säkerhetsteam som vi har i Stockholm och att de gör en review av alltihopa och försöker testa alla möjliga kryphål.

Inflicksfråga: På tal om ert säkerhetsteam i Stockholm, hur går samarbetet till med dom?

Dom jobbar ju som sagt i Stockholm så de får koden, dom har ju först en punktlista på typiska fallgropar och kända fel och hot, sen så vet jag inte exakt i detalj hur dom arbetar men dom försöker ju hitta luckor helt enkelt

Inflicksfråga: Vet du ifall de har ett ramverk som dom följer eller om det är egenutvecklat?

Det kan jag inte svara på det har jag inte koll på. Sen har vi ju förstås också testare som vi har här i Malmö så dem har ju också ett visst säkerhetstänk och kunskap så de testar också uppenbara fall. Jag menar gör vi en mobil webb så ser dem till att man inte kan injecta script tex och dom grundläggande sakerna som man förväntar sig som kan ske.

Inflicksfråga: Lägger era testare in något ”hot” alltså försöker själva skapa problem eller testar de bara att de fungerar

Det är en del av deras testprotokoll, alltså som alltid görs.

Inflicksfråga: Vet du om testarna använder sig av nåt ramverk, eller går de efter känsla eller råd och rekommendationer från er?

Dom sätter ju upp ett antal use-cases som ska testas av, vet ej om de följer ett specifikt ramverk där, men det tror jag säkert att dem gör om jag får gissa då, för de brukar vara ganska rigoröst och detaljerad testning.

När ni krypterar kommunikationen vad brukar ni göra och vad är det viktigaste?

De gångerna vi krypterat så har vi använt https, eftersom allting går via http så använder vi https då slänger vi på den krypteringen och det är ju en färdig standard så det är oftast enkelt för oss att lägga på de bitarna. Sen om vi krypterar data vi sparar på telefonen så försöker vi använda dom standardramverk som finns där även om vi tex sitter i en Android app, java har ju ett färdigt krypterings library. Javax.cypher eller vad det nu heter som vi använder. Sen är det ju intressant det här med krypterad data för att det är ju framförallt i Apple världen väldigt känsligt eftersom Apples appar säljs från USA och om jag i Europa köper en app från USA så räknas det som en export, dom exporterar en app till mig. Och om dom då ändå exporterar en app som stödjer kryptering så blir helt plötsligt dom väldigt nervösa att den här appen försöker kryptera nåt som dom är intresserade av och då måste

man fylla massa krav att man lovar att inte kryptera. Det är Apples krav och det är förmodligen den amerikanska staten som ställt till in det så. Android går runt för deras system är mer distribuerat så där har man inte samma krav. Så när man laddar upp en app så har man en fråga i App Store, om du krypterar information, ja eller nej, svarar man ja så måste man också skicka upp ett dokument som säger att man får göra det. Men när man googlar lite grann på det så är det aldrig någon som blivit kollad egentligen så de flesta svarar nog nej så är det ingen som kollar det då. Det är mycket byråkrati i USA ja.

När ni validerar input, hur tar ni er till har ni nåt ramverk som ni använder er av? Data som sätts in i appen från användaren då.

Nej där har vi väl ingen input, dom gångerna jag validerar då vet jag ju var datan kommer hamla slutligen så då gör jag oftast en enkel alltså kollar så att en email adress är en email adress och såna saker och det är ju ganska enkel check. Sen så är det upp till back-end servern där att se till att det inte är SQL-inject och såna saker. Och vilka paket dom använder kan ju inte svara på men jag vet att de använder färdiga moduler för den.

Kan du förklara en viss typ av Input validation-fel som testarna då hittar och skickar tillbaks till är eller har det också med back-end att göra.

Nej dom felen som jag validerat är att ett telefonnummer ska se ut som ett telefonnummer, en email adress ska se ut som en email adress, det är den typen av check jag gör. Sen gör jag förstås enkla saker som att om du skickar in det som en Query så ser jag till att det inte är nån olaglig kod i den, den biten.

När det gäller era kunder förväntar de sig säkerhet, en säker mobilapplikation?

Oftast är det så att dom vanliga kunderna inte efterfrågar säkerhet för dem tänker inte på det, det är sånt som vi brukar upplysa de om att vi vill lägga på den här biten och att vi vill göra det här också. Om jag däremot skulle jobba med en bank till exempel då är dom väldigt medvetna om säkerhet däremot där har dom ju många krav på säkerheten men det är ju för att dom verkar i en värld där det är så pass säkert. Oftast när jag jobbar med en kund så är det upp till oss att föreslå och ja berätta vad vi vill göra. Det är sällan så intressant för en kund, de vill hellre ha en snabb app, det brukar vara de kraven som kommer.

När ni autentiserar, exempelvis SSL, hur tar ni er till, vad använder ni er av?

Syftar du autentisering du på användare autentisering?

Ja!

Vi har inte haft så mycket inloggningskrav i de apparna jag skrivit. Den enda appen är när jag jobbat mot en bank och då har, det enda som skett är att jag har krypterat och efter det är det back-end som hanterat autentisering mot ytterligare system längre bort. Jag krypterar vår trafik däremellan men känslig autentisering sköts av back-end.

Använder ni är av några verktyg som hjälper er att skapa säkra appar?

Vi använder ju färdgiga paket för att sköta kryptering eller att sköta kommunikationen och det är ju sånt som jag sagt tidigare https det är samma sak som färdigbyggda, jag menar, på en Android app använder vi Pro-Guard för att göra den svårare att avkoda och det hjälper också upp med säkerheten men lika mycket att skydda källkoden. När det gäller Apples så är det väldigt mycket som är inbyggd i byggprocessen, hela appen krypteras ju redan, det jag skickar upp med nyckel som jag får från Apple. Där styr dem också på vilka devices jag får installerar på, installerar man inte via App Store så är det väldigt begränsat.

Testar ni hela applikationen lika mycket eller vissa delar mer än andra?

Jag vet att dom fokuserar på vissa delar av appen, speciellt när det gäller säkerhet så är det absolut viktigt att testa dom för det är två anledningar, det får ju inte gå fel, har man implementerat en säkerhetslösning så får det ej gå fel, plus att eftersom man har implementerat en säkerhetslösning så finns det mer möjlighet att det går fel, så att det är en mycket känsligare del i appen då det blir det indirekt att man kommer att testa den mycket hårdare.

Definierar ni säkerhetsmålen i början av ett projekt, eller kanske längre fram?

Säkerheten blir ju inte i början en egen punkt, det är sånt vi definierar under projektets gång när vi ser vilken del av appen som är känslig och det är alltid kommunikationen mellan applikation och back-end som när man ser att här måste vi säkra upp vi måste skydda API:et så att inte vem som helst kan komma åt det, vi ser över vilken typ av data och om vi behöver kryptering och sådana saker. Sen märker man också när det gäller projektet när man börjar optimera applikationen så börjar man cacha viss del av data och då börjar man fundera vilken typ av data man cachar och om man behöver kryptering där också. Så att det är inget vi estimerar från början men vi vet om att det kommer dyka upp.

Inflicksfråga: Kan man då dra slutsatsen att ni heller inte när ni modellerar tar upp redan i modelleringen säkerhet och hot?

Nej, har vi gjort en stor back-end så modellerar vi stresstester, den typen av säkerhet att servrarna klarar av belastningen men för apparnas perspektiv dom modellerar vi ju inte vi försöker designa på det sätt att den mesta del av säkerhet och känslig data ligger hos servern, snarare än i klienten för klienten har vi ju ingen kontroll över när den väl börjar spridas ute på många devices men servern har vi fortfarande kontroll på så vi försöker fortfarande lägga så mycket av den logiken och funktionen där.

När ni lagrar i applikationen så använder ni er av det som finns på enheten?

Precis vi följer plattformarnas egna kontexter, hur de sparar data i sina kontext, både Android och Apple har gjort det så lätt så att det är egentligen svårare att inte använda deras färdiga system.

Erbjuds det utbildningar inom företag kring säkerhet?

Nej vi har inga utbildningar inom säkerhet utan det blir ju upp till det projekt man befinner sig om man känner att man behöver nånting mer så får man se till att få in det under projektets gång. Men det är inget proaktivt så att vi går några säkerhetskurser. Behövs det tar vi oftast hjälp i första hand av säkerhetsteamet i Stockholm och då skickar vi upp ett mail och frågar exempelvis ”vi ska göra det här, vad tycker ni vi behöver till det?”.

Håller ni er uppdaterade angående hot som existerar?

Ja generellt så är vi ju alla intresserade av teknik, så vi håller oss alla uppdaterade om vad som är nytt och vad som händer inom de specifika plattformar som vi är intresserade av. Då får man ju reda på det och eftersom just jag jobbar mycket med iPhone så kollar jag upp mycket vad som händer inom iOS-plattformen. Det blir vissa tekniker som följer med att man blir intresserad av. Det är däremot nog få som följer säkerhet specifikt men vi följer som sagt de stora plattformarna och därigenom får vi reda på de stora hoten/riskerna också. Denna information kan vi exempelvis få från våra favoritbloggar som vi följer, IDG använder man först för att sålla det viktigaste, sen har jag Macrumours och The Loop som är två Apple bloggar som täcker det mesta som man behöver ha reda på därifrån.

Håller ni era verktyg som ni utvecklar med uppdaterade?

Det är ju alltid enklast när vi jobbar själva och får göra våra egna projekt för då kan vi ju alltid bestämma själva och då arbetar man alltid med de senaste verktygen. Hamnar man hos större organisationer och blir inhyrd hos en specifik kund så får man följa deras krav men absolut, det finns ingen anledning att hålla sig kvar i gammal mjukvara så vi försöker alltid hålla oss uppdaterade när vi kan.

Skiljer sig säkerhetstänket när det gäller olika plattformar?

Ja, man känner sig tryggare generellt i iOS, nu har jag bara jobbat två veckor med Android men jag känner mig tryggare generellt i iOS, från ett användarperspektiv så vet jag som utvecklare att man kan inte göra så mycket dumma saker med en iPhone app och det kan ju förstås också vara väldigt frustrerande som utvecklare ibland också. Men jag vet samtidigt om att det inte är mycket skit som kan hända där. Och jag tycker att det finns mer inbyggt i IOS som hjälper dig med säkerheten som att det blir krypterat och att det blir inlåst där i ytor där appen bara rör sig inom ett visst .

Anser du att riskerna på de olika plattformarna är desamma, eller finns det vissa som kan uppkomma på ett ställe och inte på ett annat.

Jag tror att hoten är större på Android och det mer att där kan du installera appar från ”osignerade” leverantörer, du kan installera appar hursomhelst och utgår man ifrån att en användare kommer att göra precis vad som helst så tar det inte särskild lång tid innan dom har installerat en trojan. Det är klart att man även kan få in en dålig programvara i en iPhone också men det är en större tröskel där och får du väl in en så kan du inte riktigt göra lika mycket dumheter som du kan göra med en Android app, så på det sättet är det ett större hot mot Android. Och sen är det så att Android har större spridning på enheterna eftersom du har en stor räckvidd med billiga Androidenheter till väldigt dyra så når du ut till mycket mer folk och du har då också möjligheten att nå ut till folk som kanske inte är så teknikkunniga då får man en större möjlighet till att det ska sprida sig. Så därför tror jag att Android är en attraktivare plattform att rikta in en attack mot.

Tar ni för givet att iOS miljön är så pass säker eller är ni beredda på att ta attacker?

Nej man tar aldrig nåt för givet, det är nått man lärt sig, de är så bra på att hitta på saker de där hackarna men dom få punkter jag försöker tänka på när jag utvecklar en app är dels att göra det tillräckligt svårt, nån kommer att komma igenom om dom ger sig fan på det men gör det bara tillräckligt svårt så att det inte blir värt det och när de väl kommer igenom att det dom kommer in i inte ska vara så värdefullt och det är lite det jag menade med att man ser till att mycket av det värdefulla ska ligga i back-end och inte i klienten.

Har ni förövrigt ramverk eller metoder som ni använder gemensamt, det behöver ej ha med säkerhet att göra när ni utvecklar?

Ett Cybercom projekt använder alltid en SCRUM metodik att utveckla genom så att vi enkelt och agilt kan få in nya funktioner. Vi använder ett projekthanterings- och bugghanteringsprogram som heter ”JIRA” där vi kan följa upp allting som görs i projektet. Sen så har vi också repositories, ”GIT” för det mesta nuförtiden innan var det XML. Grundbulten i ett projekt är dom tre grejerna.

När det gäller hot och risker, traditionella risker för datorer och risker för enheter?

Ja det finns vissa skillnader, en enhet är i sin natur nånting du alltid har med dig så risken är att du kommer lägga den på bordet på kafeet eller whatever och då blir du plötslig av med känslig data och sitter du på en företagstelefon som kan koppla upp sig mot ett VPN så har du kanske plötsligt en ingångspunkt till hela företagsnätverket. Du är inte lika rörlig med en PC och den är inte lika lätt att sno så därför har du där en mycket större risk vid de mobila enheterna. Samtidigt om man ser dom som nånting som man ska hacka och komma in i är det förmodligen lättare i PC eftersom de har en mycket mer öppen, det finns fler möjligheter helt enkelt. En Android eller en iOS enhet är mer låst och specificerad på exakt vad man kan göra. Nu blir ju enheterna mer multifunktionella man kan göra vadsomhelst men de är fortfarande ganska begränsade i specifika funktioner medan en laptop ska då klara av allt och de gör ju dom också mer sårbara för attacker.

Tror du säkerhet kommer bli viktigare i framtiden när det gäller mobilapplikationer?

Ja det har man visserligen sagt att det kommer bli viktigare i de senaste fem åren men ja det kommer bli viktigare eftersom vi har ökat vår mobilanvändande successivt sen iPhone introducerades så har mobila användare ökat, sen kom ju Paddorna för ett tag sen och det har ökat ytterligare. Eftersom vi använder det mer så måste vi tänka på mer säkerhet. Och det är ju på samma sätt som vi har lärt våra föräldrar till exempel att ni måste ha brandvägg och virusskydd så tror jag att man kommer att göra samma sak med mobiler inom en snar framtid.

Finns det nån mer du vill tillägga?

Nej.

7.3 Bilaga 3 - Transkribering av intervju med Softhouse

Transkribering av intervju med Petter Sandholdt, Developer and Security Consultant i Softhouse

Berätta lite om dig själv, din bakgrund
Jag jobbar som systemutvecklare/säkerhetskonsult. Vi jobbar liksom med systemutveckling och jobbar med säkerhet inom systemutveckling och jag kollar igenom det ni egentligen pratar om i uppsatsen, säkerheten men på ett mer generellt sätt för utvecklare liksom. Jag har både suttit ute hos kunder och gjort rena granskningar på kod och jag har som säkerhetsarkitekt liksom försökt i ett tidigt skede och kolla så att vi har med alla delar vi behöver för lösningen till att jobba med hur man förändrar utvecklingsprocessen till att innehålla säkra steg i utvecklingsprocessen att liksom föra in när man ska testa vad och försöka få in det så tidigt som möjligt så att det inte blir Oops, eller för den delen att man testar det för sent helt enkelt. Så det har varit mitt intresseområde och jag har kört väldigt mycket föreläsningar inom säker systemutveckling, där man liksom pratar om vad det är man ska tänka på, i vilka lägen och försöka applicera det så nära dom kursdeltagare jag haft som möjligt i deras dagliga arbete. Så det är väl vem jag är i grunden och då är det väldigt lätt att börja jobba rent, men säkerhet är ju ett så brett område, säger man ordet säkerhet på svenska så kan det betyda allt från, får jag komma in i det här rummet till är den här variabeln validerad. Folk har jättemånga ord för det och det är jättesvårt, man måste helt enkelt berätta att detta menar jag med ett hot. Så det är mycket rent utbildande för att folk ska förstå vad man pratar om. Men jag vill alltid vara tillbaka nära utvecklingsmiljön eller sitta hos utvecklingsgrupperna för att hjälpa för detta tycker jag är effektivast. Så det är jag som person och det jag har engagerad mig mycket inom är då bland annat OWASP som är Open Web Application Security Project, där man tittar lite grann på dom här sakerna, både på vilka sätt webbsidor kan attackeras och hur man kan skydda sig där. Så det är väl det jag jobbat mest med sen så har jag även gjort några granskningar enligt BC som är ett annat projekt där man egentligen jämför vilka säkerhetsaktiviteter företaget gör och graderar dom och så kan man jämföra olika företag med varandra hur bra dom är på säkerhet eller egentligen vad många företag vill är att få reda på vad det är andra gör som vi inte kanske gör så på det sättet kan det vara bra, men det blir också en jämförande skala då. Så det är väl lite grann om mig, vem jag är.

Hur Softhouse tar hänsyn till säkerheten när de utvecklar mobilapplikationer

Jag har inte varit med i många projekt där Softhouse utvecklat mobila applikationer, jag har varit med tidigare när vi jobbade mot Nokia. Hur vi tänkte på säkerheten då var väl, vi är ju ett konsultbolag i första hand så vi levererar vad kunden vill ha. Och säkerhet är något som inte efterfrågas särskilt tidigt, oftast. Oftast är det funktionalitet som man velat ha eller användbarhet, vilket brukar komma efter då, design och sen då man tittar på säkerhet, från kundens sida. Sen har vi försökt och pusha att även om detta är användbart och dom tex slipper skriva lösenord så måste vi kanske tänka på säkerhet, så vi har haft det med i vår agenda, men det vi har inte hållit på och påverkat kravet från kunden för att få in det om man säger så. Det är det dom vill ha som dem betalar för. Jag vet inte om hur det varit i de senare projekten, utan jag vet bara från tidigare år och vad jag försökt och hjälpa till med i de senare projekten. Säkerhet har aldrig varit huvudprioriterat och det håller jag helt och hållet med att det inte ska vara för att gör man ett helt säkert system så kommer ingen vilja använda det, och det är inte det som är målet. Utan säkerhet ska vara utefter den kravnivån, man ska liksom förstå vad riskerna är men sen behöver man inte ha löst dem, nä men jag menar det . Man ska liksom förstå att man kan göra det här för att komma åt någon annans form av information och tycker då att kunden eller vi och den som är ansvarig för projektet – ok den risken kan vi ta, det är ingenting vi ska fixa. Är det väldigt långsökt att vi ska hamna i fallet att det måste vara en person med en viss rättighet som just känner sig illvillig och gör det här och det här kommandot som man måste komma på att han ska göra och att det är liksom långsökt då kan man vilja att ta ner hur stor chansen är att de här hotet ska ske liksom. Så risken blir därmed liten, man har ett visst hot och så ser man, vad är sannolikheten för det? Och då säger man att risken blir låg, så det är lite grann riskberäkning, och de händer ju rätt ofta så att Softhouse har väl egentligen inte , tidigare har vi i alla fall inte haft någon uttalad säkerhetspolicy, det vi har gjort i Softhouse däremot är att jag har utbildat de flesta i säker systemutveckling så alla i alla fall förstår innebörden av många av de problem som kan uppstå, vad innebär det ifall man inte validerar saker som kommer in , vad innebär det om vi skickar SQL-anrop mot servern så man tänker på vad det innebär och vad innebär det fall man kan gå direkt till en länk utan att man validerar just för den länken. Liksom vanliga fel man gör på webbsidor, man går igenom många olika fall på hur man gör. Så jag har haft kurser med alla andra som har varit inblandade, i år har alla i vår mobila team varit med, jag har kört en stor insats med det här så alla ska i alla fall veta vad det handlar om och på det sättet kan ta välgrundade beslut på om det här är nåt man ska jobba med eller inte liksom, hjälpa till med kunden. Så det är väl det som Softhouse gjort vilket är ganska bra, vilket också har vatt liten av en ”edge” för oss – vi förstår säker systemutveckling – använd oss. Sen är det inte alltid kunden vet vad det innebär, det är inte alltid efterfrågat. Det kommer ju mer och mer men ska

man kolla lite grann på historiken så säker systemutveckling kommer ju egentligen från bankindustrin för att kreditkortsföretagen kunde ställa krav på dem som använder kreditkort och säga att om inte era system är säkra så stämmer vi er, de gjorde det enkelt för sig liksom så om det kommer ut en massa kreditkort så försöker de ta reda på hur har det kommit ut. Och har det kommit ut från ett visst system då gör dom ett – kallas det för ”Forensic”, man gör en undersökning där, tar reda på att det var deras fel och sen är det väldigt stort skadestånd. Så det var ju dom som förde in att för att få slippa betala straffavgift för varje kreditkortsköp så måste man ha utvecklat enligt vissa principer och det är det som PSI-regelverket då tog fram. Så dom drev ju det hela först, sen börja spelindustrin, e-shopparna så det är därför det har kommit och dom höll ju på pga krav som ställdes, annars förlorar man ju pengar. Medan resten av industrin har inte haft det här kravet på sig, mobilindustrin har ju inte riktigt haft det, det är ju ingen som stämmer dem ifall om man kan råka komma åt din dator över spel eller din kalender, det har liksom inte varit en stor grej för folk kan inte göra så mycket emot det. Men nu är det ju mer och mer bankärenden man gör över smarta telefoner, men det är ju inte mobiltelefontillverkarna tar ju fortfarande och säger att det är apptillverkarnas ansvar och problem så det kommer liksom på baksidan där det är inte så att enhetstillverkarna måste bry sig lika mycket. Det har ju kommit i mobiltelefonsbranschen alltså på enheterna enbart för dem vill hindra att folk jailbreakar sina telefoner och de förlorar pengar för de inte kan sälja appar och inte längre kan garantera operatörslåsning för att folk lyckas bryta sig ur det och därigenom kommer pengar förloras men det är fortfarande ca 4 procent som gör detta. Om vi gör en mobilapplikation så beror det väldigt mycket till vilken bransch vi gör den applikationen för, både vad vi tycker vi ska lägga oss på för säkringsnivå och vad kunden har för krav.

Inflicksfråga; du har hand om utbildningen av säkerhet eller du gjorde det innan, och är det bara den säkerhetsutbildningen ni har på företaget?

Ja jag kör den fortfarande, det har varit både endagars och tvådagars men det har varit fler olika varianter av den som jag har både kört internt på företaget och för våra kunder. Så vi har den externt genom Softhouse Education, det finns ett antal kunder som har tagit den då jag kört hos dem eller att dem varit här.

Tekniska Frågor

Kryptering vid kommunikation, brukar ni göra det och ”allvarliga”/seriösa är ni med det? Till exempel SSL.

Det beror litegrann på vad applikationen gör självklart, är det ren hämta information för att visa så är det enda man skyddar sig mot då är man-in-the middle attack. Dvs att nån ska påverka data som hämtas emellan och hämtar vi vädertjänst så näää, jag menar det är inte så viktigt om de lyckas visa fel väder. Om vi utvecklar mobilapplikationen så är det inte alltid vi som utvecklar back-end och utvecklar inte vi back-enden så är vi ju styrda till vad dom har gjort för möjligheter mot back-enden. Och det kan ju vara alltifrån till att dem har en helt öppen webbsida vi ska parsa eller bara hämta , till att den ska vara, de har ju varit då det varit SSL helt enkelt så vi har kunnat garantera att vi får från rätt server, om vi har rätt certifikat och att de inte byter certifikat. Det är lite jobbigare när man kör SSL då vi måste lita på deras certifikat på nåt sätt och det kan man göra genom att de har en versign certifikat, level3 eller nåt sådär, känt certifikat som vi kan lita på vilket alla mobiltelefoner har såna som standard i sig. Men är det en speciell tjänst så kanske man väljer att sätta upp egna certifikat så vi kan lita på att det exakt är dom men då måste man hantera det med att byta certifikat osv. Det finns nån tjänst där vi även har klient-certifikat så att de vet vem vi är alltså du har på nåt sätt fått ett certifikat och lagt in det i din applikation och sen så vet servern att det är just den här användaren genom certifikat, så det är lite olika, jag har varit med om båda lösningarna att vi krypterar och inte krypterar och det beror på hur farligt det är som kommer men det är också så att kryptering i sig skyddar inte identitet utan det är certifikatet som gör det och det är inbyggt i SSL så man har den möjligheten, nu blev ett långt svar, men ja det beror på.

Inflicksfråga: Hur bestämmer ni det, har ni nåt ramverk ni följer för just det?

Nej alltså det är ju tillbaka till vad kunden egentligen har för system och det som begränsar, vi har väldigt sällan så att vi kan påverka vad dom ska ändra, och tycker vi, alltså allt det här är i samråd med kunden, vi tar ju mot ett kravspec sen kan vi påverka, vi ställer ju inte krav, vi är fortfarande konsultfirma och applikationsutvecklare på det sättet. Men på alla ställen där jag vet om att vi använt känslig data på nåt sätt, det är ju när man för ut en funktionalitet från servern till att mobilen ska göra det som man vill se till att det verkligen är skyddat här, mellan enheten och servern och då har vi i alla fall – eftersom det är så enkelt att göra en enkel SSL eftersom man kan bara lita på ett standardiserat certifikat så lägger vi oss på den nivån, det enda man får berätta då är att det finns ju ett antal certifikat organisationer som blivit tagna, vad heter dem – digicrypt eller nåt, i alla fall det

var inte så länge sen en av huvudcertifikaten genererade nya certifikat åt nån kille liksom, att google.com fick… Om en av de här certifierings-organisationerna lyckats bli tagna om de lyckats generera nya certifikat så skyddar det ju inte mig då kan vem som helst låtsas vara det. Är det verkligen sådan lösning så kanske vi även har nån annan skydd på det här, men det är ju för transport som sagt, kryptering. Så jag skulle nog säga att det är hyfsat vanligt att man låter bara API:erna köra över SSL ja, jag skulle gissa på att vi alltid gör det och att det är att vi inte håller på och ”screapar” några webbsidor och ska bara visa nån information liksom.

Inflicksfråga: Vilka plattformar utvecklar ni för?

WP, Android, Nokias Symbian, Megoo och iOS. Rim, blackberry också

Hur gör ni Validering av input, följer ni några regler eller känslan ni går efter?

Där har vi kört, det är ren utbildning, folk ska helt enkelt veta att det kan vara ett problem sen är det ju oftast så är mobilapplikationer mer av ett skal, den skickar vidare informationen till en back end så att det e inte alltid vi gör all validering i mobilapplikationer utan man låter det bara skicka rakt igenom, det enda man måste kontrollera är att är det är encodat för det sätt vi skickar vidare med. Men det är inte säkert vi gör all validering, att det är rätt alltså man kan ju ha rimlighetsvalideringar, frågar man om en längd då och det skickas 4000 då kan man ju göra den rimlighetsvalideringen i enheten men oftast så väljer man bara att säga ok det här går att skicka över xml:en eller man encodar det för att skicka det via xml:en och skickar tillbaka det och låter back-end ta hand om det. Alltså det är inte alltid vi utför valideringar, det enda vi pushar på att vi ska göra är encoding för att se till ifall vi kan skicka vidare utan att det är problem. Vi har inga kontroller på att vi validerar, vi har utbildning som ser till att folk, alltså det beror helt på projekt till projekt vilket testnivå vi har att det verkligen sker.

Inflicksfråga: Kontrollerar ni då vid stora viktiga projekt?

Jag har inte varit med i alla stora viktiga projekt men vi har utbildningen som vi har gett som ska göra att folk förstår att säkerheten ska vara med i testspacen. Sen så betyder det inte att den är med och då betyder det att folk har valt det.

Testar ni er säkerheten i er mjukvara?

I vissa fall låter man kunden testa, vi är inte helt säkra på att de alltid gör det så svaret är ju egentligen Nej, vi testar inte alltid vår mjukvara. Om vi skickar vidare den till kunden vet inte vi hur väl den testas. Som utvecklare gör vi däremot alltid lågnivåtester alltså Unittester på den nivån som du utvecklar, men det är inte det som man räknar som test egentligen, test är ju det som sker efter, man kollar flödena funkar och att de är enligt spec och ibland lägger man över det på kunderna när de gör sin acceptans dvs tar emot det och då gör dom det här, är det ett litet projekt där man gör en liten app så är det lika bra att dem gör det istället för att vi ska testa den för dem, så i vissa fall nej då testar vi inte appen utan de gör dom och det är oberoende på vilken typ av tester det är. När man lämnar ifrån sig en produkt till en kund så har man alltid en acceptans att dom ska godkänna det här och betala pengarna för det och de måste på nåt sätt se att de accepterar så de gör ju alltid en acceptanstest sen vad den innehåller, det är inte säkert att det är vi som specat det utan dem som specat vad testet ska innehålla, det kan vara att vi inte vet.

När ni väl testar gör ni det som så att ni klickar runt kollar så att det funkar eller har ni nåt verktyg som testar apparna, t.ex. bombar dem för att försöka dos-attacker, denial of service?

Verktyg är lättare och köra mot serverdelen än mot klienterna eftersom en denial of service är oftast att du gör en denial of service mot servern så folk knappar inte tillräckligt snabbt så att det ska bli en denial of service i en applikation eller det blir inte riktigt en denial of service på en enhet eftersom du bara har en input om det inte är nåt utifrån beroende på vad den tar emot. Vi använder sällan verktyg för att testa mer än simulatorer för gränssnittet möjligtvis. Jag vet att vi kört det mot serverdelarna men om vi kört det mot klienterna med verktyg jag är osäker. Den enda gången jag vet att vi använt verktyg är när vi hade kravet att vi skulle ha obfuskerad kod och då har vi testat ett verktyg att när jag försökte decompilera javaklasserna att det blev obfuskerad, det var ganska enkelt men det är fortfarande ett verktyg. Oftast är det rätt så jobbigt att sätta upp testramverk för små projekt, när vi har nya miljöer hela tiden, det är svårt att prioritera att det ska byggas upp. Vi höll på att sätta upp enhetstester på Nokia projektet, testade lite grann om användargränssnittet och sånt men problemet var att Nokias ramverk ändrades rätt mycket så vi fick skriva om rätt mycket varje gång de gjorde en ny release. Det blev liksom ingen ROI på det. För Android finns det bra testverktyg, men jag har inte varit med där och använt

det för jag har inte utvecklat för Android. Men jag tror ändå att vi inte har det utan vi testar så flödet funkar vilket kan innebära, klicka eller skicka, simulera data eller så.

Använder ni er av några metodologier, processer, riktlinjer när ni utvecklar? Vi har läst att ni använder er av TDD så det vet vi.

Alltså det är också från projekt till projekt av vad som vi använder oss av men som företag ska vi använda TDD. Eller försöka använda TDD till så stor grad som vi kan. Vi ska även använda SCRUM eller motsvarande agila metoder för att få korta leveranser. Vi försöker så mycket som möjligt att alla projekt ska använda CI, Continous Integration att vi försöker bygga en ny version så fort man lämnar ifrån sig kod. Tex när du checkar in kod i vilken versionhanteringssystem man nu använder sig av så när du checkar in kod skall den testbyggas, testköras och få ett testbygge egentligen. Vi har i några projekt även gjort att den deployas, asså att testversionen så fort den har byggts hamnar på telefonen. Eller på servern. Jag vet inte om vi gjort det på mobilprojekten men jag vet att vi gjort det på serverprojekten alltså att man har den delen också vilket då kallas Continous Deployment istället för integration. Andra metodiker är att vi försöker föra in så mycket SDL (secure development lifecykle) som det går SDLC beroende vad man säger, denna är Microsofts egna som de definierat och är en väldig stor process, den agila varianten av denna har vi kört delar av. Eftersom mobilappsprojekten är så kortlivade så är det inte alltid värt att föra in alla steg utan det blir mer av en vattenfall alltså om det tar kort tid att utveckla det färdigt då är det kanske bättre att köra visa tester i slutet och sen köra om dom grejerna, fixa om dom grejerna än vad det är och försöka få in testerna ofta för det är så kortlivat projekt. Längre livade projekt så har vi ändå jobbat mer, det har gjort i större projekt jag varit med är att vi har gjort HOT/hood- modellering vilket då innebär att man kollar vad är det för typ av data som förflyttas genom systemet och hur och sen kollar man, skulle man kunna göra en man-in-the-middle attack här, har vi krypterat här, kan man låtsas vara nån annan, skulle det kunna vara att vi skickar till fel server hur vet vi det liksom, eller kollar om vi ej validerat data/input vad kan hända här. Tittar lite på hoten vid trust-boundry alltså mellan gränssnittet. Och det är också med som en del i SDLC, så vi har gjort det på de större projekten, på de små är det bara att vi ska visa nånting i appen så är det inte värt det. Vi vet vad det är som skickas vi behöver ej göra en större grej av det. Det är väldigt sällan man gör en metodik eller så fullt ut. SCRUM gör ju ingen som det exakt står i boken till exempel men man gör nån agil metod.

Inflicksfråga: När ni modellerar hur gör ni det? Står det i SDLC eller hur gör ni?

Vi brukar försöka använda oss av data flow diagrams, man kollar dataflödet mellan olika data entiteter. Ibland så ser man inte att det finns nåt mer värde att göra dom här och då gör man oftast ett sekvensdiagram dvs att det skickas en log-in från den här appen till den servern sen skickas vidare till den servern och tillbaka och då kan man använda det för då ser man ändå entiteterna som är ”där uppe” och man ser gränssnittet emellan men det går lika bra och använda en sån men om man gör data flow diagram DFD kan man även se inom enheten att först lagrar vi ner lösenordet i den här filen innan man skickar den dit och då kan man se saker som man missar om man då istället hade gjort ett sekvensdiagram. Så ja, det är dom två sätten vi gör.

Definierar ni säkerhetsmålen i början av ett projekt?

Det är oftast kunden som gör, det är ju lite kravställningen, vad dom har och om vi har lagt till nåt så ska det vara med i kravställningen, vi definierar sällan kraven på oss själva eftersom ökar vi på den massan så betyder det att det går åt mer tid, vilket betyder att det då blir dyrare och så måste ju också kunden vara med på det. Så att det görs tillsammans med kunden, vad vi har för krav på det här, kraven är ju oftast vagt definierade, t.ex. ”det ska inte gå och hacka det här”, nej men man måste ju lägga sig på de nivåerna och då är det oftast så här att man lägger sig på den här nivån att vi ska ha gjort de här och de här när det gäller tester, vi ska ha de här och de här dokumenten, artefakten eller vad man brukar kalla det. Så ja det försöker vi göra det vet jag att vi gör men det är också så att ibland kommer dom ju senare att när man väl har börjat projektet så kommer det senare ”amen du, borde vi inte tittat på det här och det här” och så får man liksom börja med det här då, det tror jag absolut för ett projekt tar ganska lång tid i diskussion innan det väl börjar så det är svårt och säga ”här börjar projektet” men på nån nivå gör vi det ja.

När ni lagrar data, brukar ni kryptera det och brukar ni tänka på var ni lagrar datan och hur ni lagrar den?

Jag tror sällan att vi lagrar med kryptering utan vi lägger vi det i ett secured storage, det finns ju i de flesta enheterna nån typ av storage du kan använda som räknas som säkert. Sen så har ju folk lyckats ta sig runt det

ibland men det är ändå det säkra som finns det finns ju nånstans där du kan lagra undan nycklar, eller saker som är bara för appen eller saker som sparas för användaren. Om vi lagrar saker själva? Det gör man inte så ofta som sagt utan man använder antigen en cache eller lagrar i den här identitetsstoragen, väljer vi att lägga det utanför tror jag knappast det krypteras för det räknar vi nog med att det inte är viktig information. Så nej.

Håller ni er uppdaterade när det gäller nya hot och nya metoder/åtgärder att bekämpa dessa och så verktygen ni använder er av?

Nej, eller det vi har sagt att du som utvecklare som jobbar i en viss miljö måste ju veta vad som gäller för den och är du Java utvecklare och jobbar du mycket med Android så måste du själv lyssna på vad som händer där, både för verktygen och för säkerheten men det är oftast tillräckligt mycket för att bara lyssna på verktygen och inte på säkerheten. Vi är några få på företaget som tagit på oss att titta på vad som händer inom säkerheten och försöka sprida den informationen vidare i organisationen, så jag tittar ju mycket på vad som händer inom säkerheten, nya hot och så där, men sen så är det inte bara inriktat på appar, vi gör ju all typ av systemutveckling så det är ju så att jag skickar det vidare till andra som jobbar med det. Och det är ju väldigt svårt, det finns ju ett forum för allt du jobbar, plattformar osv, det finns ett forum för Android, ett för Java, ett forum för varje tredjepartprodukt du använder, det är väldigt jobbigt och hålla sig uppdaterad och det är verkligen ett problem att vara det så förväntningarna ligger mer på utvecklare som person än att vi i projektet tar reda på det men det är ju också så lite grann om att approjekten är så kortlivade, ett approjekt är oftast under ett halvår och vad som hänt i branschen under det halvår är inte så mycket, visst kan det under tiden ha kommit nåt men man brukar nog försöka hålla sig uppdaterad i början när man väljer vilket vilket ramverk ska vi använda, vilka verktyg vi ska använda, vilka tredjeparter ska vi använda, det försöker man bestämma så tidigt som möjligt och det är då man försöker hålla sig uppdaterad. Jag skulle gärna se att man gör det när man väl valt det, verkligen, enligt SDLC så ska man ju alltid definiera det, skriva ner vilka tredjeparts du valt, så att man lätt kan se om det kommit nya versioner och varför, liksom bara kolla på dessa verktyg och se här och där. Och det är vi nog lite dåliga på att skriva ner vilka vi har och sen är det också det att väljer du en tredjepartsprodukt så kan den innehålla fler under sig och ja hur ska du hantera det då? Ska du uppdatera dem i det här eller ja, det kan ju bli rätt rörigt. Så att nej det tror jag inte vi är särskilt bra på.

Infliksfråga: Anser ni att hoten och riskerna är detsamma för alla plattformar, men att kanske en plattform är bättre än någon annan på att skydda mot ett visst hot?

Det är fortfarande samma typ av hot som gäller och det som man var rädd för var maskar och liknande som hade möjligheter och där hade Android, det var ju lättare att göra en mask för Android än för iOS för att folk kunde Java och inte Cocoa, det är ju så enkelt egentligen. Men kollar man på mängden maskar som utvecklats för mobilerna så har det blivit, det finns ju tiotusentals kanske men på senaste året har de kommit ca två procent till, liksom det är en väldigt väldig långsam utveckling av nya maskar, kollar man på F-secure eller på Mcafees lista över kända maskar så är de listorna, de har inte ökat särskilt mycket och det är ju fortfarande samma maskar som fanns i Java eller runt java-applikationer tidigare. Just maskar var man verkligen mer orolig för än vad det verkligen har slagit ut om man tittar på statistiken även om det var kinesiska säkerhetsföretag som började skrika här nu nyss oj det är jättemycket maskar här nu, vi har hittat fyratusen procent mer maskar och det stämde ju i deras fall för de jämförde ju med förra året när dem inte fanns, dvs företaget fanns inte så de hade ju inget att jämföra med, deras databas var jätteliten och ny och så sa de ”oj nu har det ökat jättemycket” för nu hade de börjat söka liksom. Så för dem så ökade det fyratusen medan för andra ökade det med en-två procent, det har inte ökat så mycket sa de andra säkerhetsföretagen. Så det är farligt de här procenten som dessa presenterar liksom. För det var fortfarande tolv eller så där. Andra hot som finns om det inte är maskar som egentligen är det som utnyttjar plattformen mest då är det ju phising-attacker och det är key-loggers, det är inte så mycket mer i nån av plattformarna egentligen.

Inflicksfråga: Så det är helt enkelt samma hot oberoende plattformar?

Ja det är så jag uppfattar det, varför t.ex. Meego ej attackeras mycket eller Rim för att dessa inte används lika mycket som exempelvis Android och iOS så det blir mycket färre att bråka med och då är det inte lika roligt och intressant så det är ju Android och iOS man vill bråka med för de säljer mest och används av väldigt många människor runt om i världen och Android har väl haft nackdelen eftersom de inte haft sin process för nya appar lika kontrollerad som den som Apple kontrollerar för iOS så har det kommit en massa attacker och phishing appar och andra låtsas-appar in i Android som inte kom till iOS för att de har mer kontroll så det har inte kommit lika mycket fejkade appar där, så det är väl en skillnad så där. Men sen vet ju android-användarna om det här och är väldigt mycket mer restriktiva på vad dem installerar, alltså dom som tänker på detta medan på iOS installerar

man bara alla spel man hittar så att utslagningen blir ungefär lika högt för det är en annan typ av användare för att i iOS så tror man att man är säkrare pga att det är Mac och de har det ryktet medan i Android är annorlunda och är Linux, och Windows Phone det vet jag inte så mycket, jag har faktiskt ingen koll på WP det har inte funnits så mycket, det finns ju potential att det kan komma eftersom det finns så många färdiga kit för att skriva virus till Windows att det kanske skulle kunna slå hårdare där men då måste det komma in i telefonen ändå på nåt sätt.

Tänker ni annorlunda kring säkerhet beroende till vilken plattform ni utvecklar för, och då tänker vi främst på Android, iOS och WP?

Där får väl jag säga att jag inte vet, utifrån vad vi har sagt hur vi ska jobba så är svaret nej, för vi vill ju fortfarande köra säker systemutveckling oberoende plattform, samma problem kan finnas. Pratar vi kommunikation alltså typ SSL eller pratar vi… input validering så är det exakt samma sak oberoende plattform. En anledning till att vi skulle välja olika är för att det kanske är svårare att göra nån typ av koll eller sådär men det är upp till utvecklaren sen. Enligt mig borde vi inte ha annorlunda teori liksom det vi säger att vi ska göra, kontroller och sådär beroende på plattform, jag ser ingen anledning till det, det är inte så att den ena plattformer är så mycket mer osäker därför skulle vi ha mer kontroller, det är ingenting just vi kan säga utan enligt mig borde vi som sagt inte ha det. Sen så är det inte så många projekt som äger rum att vi utvecklar för flera plattformar samtidigt så jag vet faktiskt inte.

Tror du att ni kommer utveckla mer säkra appar i framtiden och om appar kommer kräva mer säkerhet?

Absolut tror jag att appar kommer att kräva mer säkerhet framöver, jag tror att all mjukvara kommer att kräva mer säkerhet framöver eftersom folk börjar se mer och mer vad som kan gå åt skogen. Det är mer och mer identitetsstölder och mer och mer sånt som händer och det finns ingenting som säger att hotet sitter i huset bredvid det kan lika gärna vara i en annan världsdel för vi är så ihopkopplade tack vare webben. Så att jag tror absolut att fler och fler kommer att se behovet på nåt sätt av att lägga in fler kontroller och det är ju också det att mobiltelefons branschens appar har ju många av dem varit grattis, på iOS har du fortfarande en stor del appar som kostar men det är fortfarande spottstyver och kostnaderna brukar inte finnas i apparna så mycket, det man däremot rör sig mot är mikroekonomin att man alltså tar betalt för små saker i själva apparna och då börjar kraven komma mer i apparna för att när man ska ta betalt för att köpa appen då är ju problemet hos Apples App Store eller Androids Google Play medan om man gör mikrobetalningar, antigen att man använder det färdiga ramverket i plattformen, det är ju inte alltid man gör så, ibland tar man betalt på andra sätt. Exempel att man köper saker via nån annans betalningsportal och då kommer kraven närmare appen så jag tror att eftersom vi utvecklas till att kunna göra mer och mer med våra telefoner att kunna betaltjänster inne i apparna så kommer det vara mer kravställda utifrån kunderna s jag tror absolut att det kommer komma mer och mer eller behöva mer och mer. Man är inte så rädd om sin identitet och sin information, alla skriver och uppdaterar om vad de gör redan på facebook idag och har gjort det ett bra tag, folk är inte så rädda och berätta vad som händer i deras liv eller vad dem gör eller nåt så där så det viktigaste kommer tillbaka till pengar ändå för det verkar vara det som folk är känsligast för.

7.4 Bilaga 4 – Transkribering av intervju med Anonym

Transkribering av intervju med Anonym, Business Manager på Anonym.

Berätta lite om dig själv, din bakgrund?

Jag är själv Systemvetare, ja jag pluggade systemvetenskap i 4 år, fokuserade rätt mycket på extrakurser på programmering för jag tyckte att det var för lite programmering på systemvetenskap. Har jobbat sen 96 som utvecklare, projektledare, integrationer, mycket webbaserade grejer fram till 2008 jobbade jag som konsult kan man säga. Sen tog jag över mobilutvecklingen här nere.

Vilka plattformar utvecklar ni till?

Android, iOS och Windows Phone, vi har haft Blackberry innan också men inte nu längre.

Beskriv hur ni tar hänsyn till säkerheten när ni utvecklar, vad är det viktigaste för er?

Alltid utifrån vad kundens behov är från början, vad det är för typ av applikation som ska utvecklas. Sen så finns det ju alltid minsta möjliga säkerhetsnivåer man måste ha när man ska gömma vissa grejer och sånt så att man inte kan sniffa upp trafiken och ha lätt för att få fram grejer. Ett exempel på det, är en kund som gick till en reklambyrå en väldigt känd byrå och fick en app och efter en dag så läckte den personuppgifter från telefonen, den skickade ut en massa grejer, så det är ju det minsta man måste ta hänsyn till när man gör publika applikationer då så att säga. Men vi jobbar nästan uteslutande med mobilintegration, dvs nä man jobbar in mot företagssystem, SAP eller affärssytem eller andra grejer och då är det ju jätteviktigt med säkerhet.

Krypterar ni kommunikationen?

Ja

Hur gör ni det, vad använder ni?

Ja det gör vi, HTTPS, SSL-kryptering men ofta är inte den tillräcklig och då krypterar vi datan själva innan den lämnar beroende på vad det är för lösning.

Vad använder ni er av då när ni gör det själva?

AES-kryptering (Advanced Encryption Standard) som vi har som mest idag där vi jobbar, framförallt är det för att det inte ska ligga nåt okrypterat på servern.

Validerar ni input, hur isåfall?

Ja det gör vi, vi validerar all input men jag kan inte säga exakt hur.

Autentisierar ni, vet du hur ni gör det?

Ja det gör vi, det beror på, på vissa lösningar har vi device authentication och user authentication.

Inflicksfråga: Använder ni egna autentiseringar eller existerande?

Det är existerande autentiseringar alltså det som är inbyggt i .NET-ramverket osv.

Testar ni er mjukvara?

Ja det gör vi men alldeles för lite . Vi har två stycken dedikerade testare som sitter härinne på ungefär... har vi en testare på fyra utvecklare eller nåt sånt och det är för lite så vi kommer att starta upp mer test både bland dom anställda och outsourcande-tester.

Inflicksfråga: Vet du hur dom testar, är det lågnivå tester, exempelvis Unit-tester?

Dom gör inte Unit tester dom kanske följer upp Unit-testerna men det är från projekt till projekt. Vissa projekt har du Unit-tester i koden i vissa har du inte. Och sen testar vi ju, dom är ju med i planeringen så man bygger upp sina testfall.

Använder ni er av några verktyg som hjälper er att förbättra säkerheten?

Vi har tagit in säkerhetsbolag som har validerat vissa av tjänsterna som vi har utvecklat.

Har ni några metodologier, processer eller riktlinjer som ni följer när ni programmerar?

Ja det har vi i utvecklingsprojekten, till exempel SCRUM men det är lite grann "SCRUMISH" vi kör faktiskt lite vattenfall men vi exekverar det som SCRUM-projekt så att säga.

Hur tror du att dessa metoder du precis gav exempel på hjälper er med säkerheten?

Hur just det hjälper med säkerheten är en bra fråga, alltså vi återanvänder ju en del av koden vi har så att säga, så vi har vissa komponenter som vi redan vet är säkra som man kan dra in i projekt men det är nog inte den modellen som avgör om nåt blir säkert eller inte men däremot så kanske det blir lite mer agilt och kan prioritera om saker men det förbättrar kanske inte projektleveransen och säkerheten för det.

I början av era projekt, definierar ni säkerhetsmålen?

Ja, är detta ett projekt, en produkt eller en lösning som bör vara säker så gör vi det. Men är det inte det så har vi basic standard form av säkerhet.

Inflicksfråga: Känner du att kunder efterfrågar säkerhet?

Ja, vi jobbar nästan uteslutande kunder som efterfrågar det mångt och mycket just nu och det är för att vi jobbar med mobilintegration, alltså vi integrerar ju mot deras system och därför ska vi få bevisa att det är säkert att göra det och då räcker det ju inte med SSL-hanteringen, du måste bland annat ha datan krypterad, basic verktyg är helt enkelt inte tillräckliga att förlita sig på. För många kunder är det som sagt väldigt viktigt, jag menar vi jobbar rätt mycket med säkerhetsbolag alltså inte att vi bara anlitar mjukvarusäkerhetsbolag för vår utveckling utan säkerhetsbolag att koppla upp larmsystem, kameror, passagesystem, och sen så även rätt mycket bank och finans, försäkring, styrelser, för att hantera deras känsliga information och rätt mycket med fastigheter liksom arbetsramar, lägga beställningar, mikroekonomiska transaktioner, så det är jättekänsligt för dom.

När ni lagrar data i appen, hur tar ni er till för att göra säkert?

Behövs det vara säkert så är det krypterat redan i appen, vi har ju allt krypterat även om vi exempelvis gör en request så får vi aldrig ner nån data som inte är krypterat. Även om vi använder SSL, så är SSL inte tillräckligt säkert.

Använder ni er av enheternas egna säkerhetsmöjligheter?

Nej, mycket på grund av att det är första som någon hackar och attackerar så är det det. Ser man exempel på iOS så är keychains det första de hackar, det tar lite längre tid att göra men det finns ju alltid nån video på YouTube som visar hur man gör.

Brukar ni kartlägga hoten i början av ett projekt genom att rita upp modeller?

Alltså vi har ju då som sagt en lösning för att distribuera saker säkert så att säga, och den modellen utgick från mer en övergripande konceptuell beskrivning av hur och var hoten finns och den omfattar i princip allt så vill vi ha nånting säkert så kör vi på den modellen.

Inflicksfråga: Så den har ni för alla projekt?

Nej utan det är mest när det krävs, den har vi liggandes, den är bara för en plattform, det är det som är problemet och just på grund när du vill hantera saker säkert på devicen så finns det ju en plattform som är mer homogen än Android för där implementerar dom beroende på vilken hårdvarutillverkare det är så lägger dom säkerhetsnivån

på olika ställen eftersom det är ett open-source operativsystem. Så på grund av att det beror som sagt på hårdvaran så går det inte att säga att just detta är säkert på Android till exempel.

När det gäller utbildning inom säkerhet, erbjuds de anställda här?

Ja om det behövs.

Inflicksfråga: Vilka utbildningar är det?

Vi har inte gått särskilt mycket utbildningar när det kommer till device-biten, vi har väldigt erfarna konsulter så det ger inte så mycket för dom att gå på dom utbildningarna. Utan mer håller sig påläsa. Däremot på serversidan händer det ju väldigt mycket hela tiden och vi går mer och mer från Java till .Net på serversidan och där finns ju mycket.

Är utvecklarna medvetna om vilka risker som existerar och håller dom sig uppdaterade?

Nu har vi lite nyanställda så jag vet inte riktigt, men dom flesta gör det, alla våra tech-leads vet. Detta gör dom genom att exempelvis läsa olika forum.

Håller ni era verktyg som ni använder uppdaterade?

Ja.

Hur skiljer sig er säkerhetstänk mellan de tre plattformarna?

Alltså vi utvecklar inte särskilt mycket till Windows Phone, det var längesen vi hade ett Windows Phone-projekt men det kommer nog snart. Hur det skiljer sig? Alltså det är på den nivån att vi har ju valt bort en plattform som jag sa tidigare på grund av att vi inte kan erbjuda säker lösning för tex Android, vi hade kunnat erbjuda det för HTC-Android till exempel men inte bara Android i dess grundutförande. Det är hårdvarutillverkarna som styr när det gäller Android så det blir lite grann där vi känner, funkar det eller inte. Sen har vi inte haft behovet av att göra riktigt säkra lösningar för Android-plattformen liksom ingen har efterfrågat det, däremot så kommer det antagligen komma för Windows.

Inflicksfråga: Varför går det så att det inte går för vissa plattformar men går för andra?

Tittar du på Apples iOS plattform så är den mer homogen, bakåtkompatibel, och bara en hårdvarutillverkare alltså det är verkligen bra ihopsat. Den är mycket lättare att göra en säker lösning på. Eftersom du vet hela tiden liksom var nånstans Apple implementerat säkerhetslogiken i plattformen. Du har inte samma sak i Android för det för där är upp till varje hårdvarutillverkare att bestämma var i lagringen eller var i stacken som dom ska implementera sin säkerhetslösning.

Upplever du att det finns nån skillnad i hoten mellan standard desktopapplikationer och mobilapplikationer, exempelvis att man-in-the-middle attack är samma?

Ja det är ju liknande hot helt klart, jaja absolut. Man hanterar man-in-the-middle genom att ha signerad mjukvara på båda sidor och signerar requests som görs och vi har autentisiering i varje anrop neråt i anropstacken så att säga på serversidan att hela tiden så blir det en autentisering så det inte ska gå att kunna komma in. Det är samma funktionalitet som jag ser det, om man till exempel kopplar upp en laptop mot wifi eller en Android telefon. Däremot så finns det ju mer sniff och sånt här på en mobil enhet.

Inflicksfråga: Den färdiga modellen som ligger liggande, vad är den baserad på?

Det är en lösning som vi har som vi säljer, en färdig paketerad tjänst liksom och det är för att vi ska kunna hosta en kunds alla dokument. För det är en står grej med mobilitet just nu som man pratar om, molnet liksom, molntjänsterna och hur säkert eller inte säkert det är. För att vi skulle kunna mitera det så sa vi bara att vi inte inte vill ha nånting i våra system, på våra serverhallar som inte redan är krypterat. För annars så har du alltid en systemadmin som kan komma åt grejerna. Många konkurrerande lösningar som har varit totalhackade för dom hackar serversidan istället. Och så har dom gjort att man “vi måste kunna komma åt det från datorn , iPaden,

telefonen och ja allt" så då gör man en webblösning och så fort du gör en webblösning då per default kan man säga direkt att nej då är det inte säkert längre, då är det osäkert.

Inflicksfråga: Den färdiga lösningen ni har, är det så att ni plockar från den i varje projekt?

Nej, det är det inte eller ja, delvis krypteringsbitarna kan jag plocka ut för att använda i andra projekt men det är inte alltid vi använder hela den infrastrukturen för att göra det för den är ganska stor och komplex. Men vi kan plocka delar från den, alltså godbitarna absolut.

Inflicksfråga: Skulle du kunna då säga att det funkar som nåt, inte ramverk, men nåt som ni följer när ni sätter upp ett projekt när det gäller säkerheten?

Det finns ju med i bakhuvudet på alla utvecklarna som varit med i det projektet så att säga men sen så har vi alltid en tech-lead som sagt som kan sitt, vi har haft med en arkitekt som förstår hela den biten.

När ni definierar projektet i början, när det gäller säkerheten, har ni nåt som ni följer eller är det mest att ni anpassar er efter kunderna?

Vi har ju som så att man många gånger följer det som kunderna vill ha så att säga. Men vi hade ett exempel nu med att det var en befintlig kund som vi utvecklat andra mobilintegrationer till och då hade dom skrivit in ordet säkerhet i sin kravspecifikation och då blev dom chockade när dom fick tillbaka offerterna från alla leverantörer dom hade skickat ut till. För den offerten var bra mycket större kostnadsmässigt än vad den varit tidigare, tidsramen var också annorlunda. Så så fort det står nåt krav på säkerhet då går man in i nån annan mode liksom, man börjar analysera på ett helt annat sätt och då är man ganska kritisk till kravspecifikation som kommer in för oftast sitter det nån som kanske kan webb och han har normala administrativa lösningar, inne på ett företag som har fått skriva kravspecifikationen och börjar man då diskutera säkerheten då kommer inte han veta så mycket, då händer det saker liksom, då börjar vi verkligen bli kritiska till det, dom har inte kompetensen att skriva ett kravspec med fokus på säkerhet, det är väldigt sällan man träffat på.

Inflicksfråga: Vad gör ni då?

Då går vi tillbaka och säger bara att detta är antigen för oklart, vi måste lägga mer tid på detta eller att enligt era krav så stämmer det inte, ni får ingen säker lösning på dom här kraven.

Inflicksfråga: Ni gör detta då efter utvecklarnas kompetens och känsla, eller?

Ja, precis.

Inflicksfråga: Det var det vi ville få fram om ni har en lista eller att det mer är att utvecklarna själva avgör det?

Men visst, det finns nån lista nånstans i systemet som bara säger tänk på detta men all mobilitet handlar om säkerhet.

Upplever ni där att det är samma hot som gäller för alla plattformarna, fast tex att iOS är säkrare?

Jag ska inte säga att den är säkrare men den är lättare att bygga en säker lösning på. Men plattformen i sig behöver inte vara säkrare i sig men det är en homogen plattform därför är det lättare att skriva en säker lösning för den.

Inflicksfråga: Så det är egentligen så att det är snarlika hot plattformarna kan utsättas för förutom att nån är bättre rustad för?

Ja, Windows Phone kommer nog vara lite mer lik Android på det viset att du har olika hårdvarutillverkare så att säga, men däremot är det inte lika, tror jag spretig som Android.

Tror du att säkerhet kommer att vara viktigare i framtiden när det gäller mobilapplikationer?

Ja, det har redan blivit det, jag tror att det kommer fortsätta växa, allting pekar på det, tittar man bara på vad alla MDM-verktygen (Mobile Device Management) klarar av idag och vad dom vill att dom ska klara av idag. Dom tar till exempel fram SDK för utvecklare så att man ska koppla upp sig mot deras MDM-verktyg och då blir lösningen fast vid deras MDM-verktyg och så kan inte kunden byta MDM-verktyg, ja. Vi har tagit det i ett steg längre, vi har använt MDM-verktyg så vi kan kryptera lagringsytan på hela telefonen. Har man väl fått upp den så ligger allt helt öppet. I vår krypteringslösning så krypterar vi dom enskilda grejerna med olika nycklar hela tiden så vi använder AES-kryptering och RSA-kryptering för att kunna distribuera grejerna. Så ja det händer jättemycket och det kommer bara explodera.

Har du nåt du vill tillägga inom vårt ämnesområde?

Allt behöver kanske inte va säkert heller, men det finns alltid den miniminivån som man måste hålla, det spelar ingen roll ifall det bara är den publika som är lite, typ reklam eller vad det det nu kan vara så att säga, så kan vara en tjänst så att man har denna nivån i alla fall.

Då tackar vi så mycket för oss och att du tog dig tid att svara på våra frågor!

Tack själva och hoppas det gav nåt och lycka till!

7.5 Bilaga 5 - Transkribering av intervju med Xdin

Transkribering av intervju med Björn Nilsson, utvecklare på Xdin.

Om du kan berätta lite om dig själv, din bakgrund

Pluggat data på LTH, därefter börja jag jobba på Axis, jobba där i ett par år, sen så börja jag jobba här på Enea, som sen blev uppköpt av Xdin, där har jag suttit en del med mobilappar, pillat med det, sen har jag suttit ute på ett företag som sysslar med fyrdjulsdrift på bilar sen nu så sitter jag med Android- och Windows 8 appar och liknande.

Vilka plattformar utvecklar ni till?

Windows 8, Android och iOS, men inte så mycket i iOS.

Om du skulle kunna beskriva övergripande hur ni tar hänsyn till säkerhet när ni utvecklar mobilapplikationer?

Det handlar ju mycket om sunt förnuft men sen så är det också beroende på kunder om dom kräver vissa grejer så måste man ta hänsyn till det för i slutändan så är det mer eller mindre bara kunden som sätter krav, om dom inte sätter krav så, visst man gör inte dumma grejer bara för det utan det är just det sunt förnuft som gäller, men det är som sagt beroende på kunderna och deras kravspec.

Inflicksfråga: Känner ni att era kunder efterfrågar säkerhet?

Både ja och nej, i dom projekt som jag suttit i då har det varit mycket prat om det i början men ju längre tiden går så kan en de kunder tycka, att "det där vi lägger tid på är inget vi får ut av" så där finns risk att man kan tumma på det.

Brukar ni kryptera kommunikation?

Det handlar väldigt mycket beroende på vad det är, men ja när man kör mellan mobilappar emellan så är det HTTPS, nån sorts av kryptering, helt enkelt.

Inflicksfråga: Hur bestämmer ni vad som ska krypteras, är det kunden eller?

När man kör över nätet så blir det ofta, alltså man kör ju väl mer eller mindre en praxis att så mycket som möjligt bör krypteras men i slutändan så är är det kunden som bestämmer om allting ska gå så snabbt som möjligt och det ska vara så lite belastning som möjligt så får man tumma på lite grejer kanske, men överlag så är det, det som går att kryptera vill man ändå kryptera

Brukar ni validera input?

Ja,

Inflicksfråga: Hur gör ni det, har ni nåt ramverk eller känsla ni går efter?

Ja det är väl mer känsla isåfall att, det beror lite på projekt och projekt, hur det sätts upp, ibland kan man från kunden också att där ska det göras osv.

Inflicksfråga: Så det är om kunden kräver det?

Ja.

Autentisering, brukar ni använd er av det?

Ja, det vill man ju ha

Inflicksfråga: Hur använder ni er av det?

Menar du alltså, hur...?

Inflicksfråga: På vilket sätt autentiserar ni att applikationen är den den utger sig för att vara eller att den som kommunicerar med applikationen?

Ja det finns ju flera olika, man kan ju exempelvis ha signaturer som man arbetar med.

Inflicksfråga: I koden eller?

Ja man kan ju signera koden och sen så den datan som skickas till den måste också vara signerad på ett visst sätt för att man ska ha ett...ja ett godkännande helt enkelt.

Testar ni er mjukvara och hur isåfall, på vilka sätt?

Ja det gör vi, genom att man försöker ha automatiska testfall ju, men lite "happy-testing" sker liksom alltid att man försöker få fram alltså testar det här och det här av sig självt.

Inflicksfråga: Är det bara lågnivåtester eller fler och mer omfattande tester?

Vissa projekt är det ju mer att kunderna vill betala alltså... som ett konsultbolag så vill ju kunderna betala så lite som möjligt och därför i vissa projekt så kanske tycker dem "amen testa? Vi vill ha koden, vi vill att ni ska fixa problemet" och då tycker dom, "test njae, det vill man inte betala för" så då kan det hamna på efterkälken och då är det mer att kunden isåfall själva testar av det.

Brukar ni testa säkerheten i mobilapplikationerna?

Njae, jag har inte varit med om det i alla fall.

När ni testar, har ni några metoder eller riktlinjer som ni följer eller är det också känsla ni går på?

Känsla och...alltså man skapar ju sig ändå en, inne på varje företag har man liksom en "så här bör man göra" men det är också till kunder om dom har sina egna typ "det här måste följas, så här jobbar vi". Då följer man deras sätt att jobba.

Inflicksfråga: Så ni har era standarder men ni kan också gå efter kundernas egna?

Fast vi har inga egna uppskrivna, riktigt standarder att så här ska det göras utan mer att man har arbetat så här hela tiden så därför har man kommit in i det flytet. Sunt förnuft helt enkelt.

Använder ni er av några utvecklingsverktyg som direkt påverkar säkerheten?

Nej, det kan jag inte säga att vi gör. Men det är samma sak där liksom, om man till exempel kommer ut till en kund så kanske dom har verktyg man måste köra då, exempelvis testa av det och liknande då gör ju man det.

Har ni några metoder eller riktlinjer eller processer som ni normalt följer när ni utvecklar mobilapplikationerna?

Ingenting nerskrivet riktigt utan det håller man på att arbeta fram men det är just det att alla kunderna vill ha det lite olika så är det rätt svårt men det är väl när man får in nåt inhasat, man får leverera en lösning, då är det lättare och sätta upp. Men vi har inte jättemånga såna.

Definierar ni säkerhetsmål i början av projektet?

Ja tillsammans med alla krav så definierar man även vilken säkerhet eller vad appen ska klara, beroende på produkt.

Inflicksfråga: Detta gör ni alltså tillsammans med resten av specifikationerna?

Precis.

Inflicksfråga: Har ni då en lista på mål som ska nås eller?

Ja, det blir väl att man sätter upp en lista på det och sen så är det ju kunden som bestämmer. Vissa mål tar längre tid, kostar mer och då finns risken att kunderna känner att man kanske inte känner att det är värt att välja det. För vissa grejer som man försöker täppa till kan vara kanske extremfall.

Kartlägger ni möjliga hot som finns i början av projektet, att ni ritar upp några modeller eller nåt sånt?

Nej, har ej varit med att man gått in så djupt.

När ni lagrar data, hur tar ni er till då?

Man lagrar det i en databas, fast det kan vara SQL-Lite databas, det är det vanligaste.

Inflicksfråga: Hur ser ni till att datan i appen är säker, eller ligger säkert?

Där litar man väl en del på som i Androids fall till exempel så litar man på operativsystemet och det som finns i enheten.

Erbjuds dom anställda utbildning inom säkerhet?

Nej.

Håller ni er uppdaterade om vilka risker och hot som existerar?

Ja det försöker man ju göra, men det är mycket upp till var och en alltså vad man har för intresse, om man nu är intresserad av Android så läser man på och håller sig uppdaterad på det, genom forum, man följer olika forum och liknande.

Håller ni era utvecklingsverktyg uppdaterade, exempelvis de senaste verisonerna?

Nja, vissa grejer, när man exempelvis utvecklar till Android, alltså utvecklar "hela alltet" så säger dom "detta är testat för Ubuntu, T04, denna version, GSS versionen och sånt" medan det kommit senast efter det, men då är det inte säkert att grejerna bygger på det och då får man ofta gå till exakt den som dom har testat den på. Så det blir inte alltid det senaste.

Hur skiljer sig ert säkerhetstänk mellan de olika plattformarna?

Ja alltså mycket gäller att man måste lita på operativsystemet, är det en rootad/jailbreakad telefon så kommer man åt det ändå, visst man hade kunnat kryptera all data, men är den rootad så kan man isåfall komma runt det och kanske få upp nyckeln. Så i grund och botten måste man lita på operativsystemet, och det är nog samma för alla plattformarna man utvecklar för.

Inflicksfråga: Upplever du att hoten är desamma mot alla tre plattformarna, fast kanske någon plattform exempelvis är bättre än nån annan på att skydda sig mot ett visst hot osv?

Jag känner nog mer att just när det gäller Apples iOS där vävs man in i en "falsk säkerhet" att där finns ju ingen hot osv, alltså på den nivån, men sen så har det ändå kommit en del rätt så stora allvarliga grejer så man bara tänker "oj var det lite problem med denna också". Medan Android, tack vare att den är så pass öppen så upptäcks problemen där och då kommer den informationen ut medan det förmodligen är många problem hos iOS, men att de hålls inne i och med att det bara är dom som utvecklar sin Operativsystem så då är förmodligen en del problem där som dom har fixat. När det gäller Windows Phone har dom fördelen att det är Microsoft, du programmerar på samma sätt som du gör för Windows och då kanske man inte gör misstagen som man kanske råkar göra när man utvecklar för Android eller iOS. Så det är en styrka för att man vet vad det är, med Microsoft osv.

När det gäller traditionella hot och dom som finns i mobila enheter, anser du att hoten är dom samma eller är dom annorlunda?

Det som jag själv känner är väl att det är exakt samma hot, det är kanske inte lika mycket virus för att skada, fast det finns väl knappt det väl längre till PC, nu är det mer hijacking, kom över grejer, phisning och det finns ju samma för de olika plattformarna, det är min uppfattning.

Är det så att ni anpassar er mycket åt vad kunden vill?

Ja det är dom som bestämmer, men visst man får ändå stå på sig om dom försöker få igenom något "dumt" så försöker man ändå stå på sig och få dom att hålla emot.

Hur står ni på er, är det från fall till fall eller har ni något ni följer inom företaget?

Mycket är ju, om dom vill att man ska göra på nåt sätt så gör man på deras sätt, det är mer av en policy att kunden alltid har rätt. Men om dom försöker ta något beslut som kanske påverkar slutprodukten negativt så försöker man ändå påvisa att det kanske är bättre om man gör såhär istället eller så. Man får ge råd, men kunden har alltid rätt så det gäller ändå att det mesta dom vill det måste man ändå göra liksom, men man får ändå ge rekommendationer.

Vilka typer av kunder har ni haft och vilka typer av appar har ni utvecklat?

Malmö Stad-appen och vi har gjort för Lugi, vi sitter också hos Sony och håller på hos dom också.

Har du nåt du vill tillägga?

Nej.

Tror du personligen att det i framtiden kommer att bli viktigare när det gäller säkerhet vid mobilapplikationer?

Jag tror att det kommer vara viktigare på en Mobiltelefon än på en PC eftersom mobilen har du alltid på dig och exempelvis att man-in-the-middle attacker då blir lättare att göra på mobiler än på en PC.

8 Referenser

Basavala, S.R., Kumar, N., Agarrwal, A., (2013): *Mobile Applications -Vulnerability Assessment Through the Static and Dynamic Analysis.* Conference on Advances in Communication and Control Systems 2013 (CAC2S 2013).

Becher, M., Freiling, F.C., Hoffmann, J., Holz, T., Uellenbeck, S., Wolf, C., (2011): *Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices.* IEEE Symposium on Security & Privacy (SP), 2011, p96-111, Berkeley, CA, US, 22-25 May.

Dunham, K., (2008): *Mobile Malware attacks and Defense*. Elsevier. Inc. Burlington, MA, USA. ISBN: 978-1-59749-298-0.

Dwivedi, H., Clark, C., Thiel, D., (2010): *Mobile Application Security.* McGraw-Hill, USA. ISBN: 978-0-07-163357-4.

Elfattah, M.M.A., Youssif , A.A.A., Ahmed, E.S (2011): *Handsets Malware Threats and Facing Techniques.* (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 12, 2011.

Fried, S., (2010): *Mobile Device Security: A Comprehensive Guide to Securing Your Information in a Moving World.* Auerbach Publications, USA. ISBN: 978-1439820162.

F-Secure (2013): *Mobile Threat Report*. Hämtat från F-Secure: http://www.f-secure.com/static/doc/labs_global/Research/Mobile_Threat_Report_Q1_2013.pdf den 20 April 2013

Gartner (2013): *Gartner says worldwide mobile phones sales declined 1.7 percent in 2012.* Hämtat från Gartner: http://www.gartner.com/newsroom/id/2335616 den 16 April 2013

Gunasekera, S., (2012): *Android Apps Security.* Apress, USA. ISBN: 978-1430240624.

Hoog, A., (2011): Android Forensics: *Investigation, Analysis and Mobile Security for Google Android.* Syngress, USA. ISBN: 978-1597496513.

Hoog, A., Strzempka, K., (2011*): iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices*. Syngress, USA. ISBN: 978-1597496599.

Howard, M., LeBlanc, D., (2002): *Writing secure code, second edition*. Microsoft Press, Redmond, Washington, USA.

Jacobsen, D.I., (2002): *Vad, hur och varför? Om metodval i företagsekonomi och andra samhällsvetenskapliga ämnen*. Studentlitteratur, Lund. ISBN: 91-44-04096-2.

Kadrich, M., (2007): *Endpoint Security.* Addison-Wesley Professional, USA. ISBN: 978-0321436955.

La Polla, M., Martinelli, F., Sgandurra, D., (2012): *A Survey on Security for Mobile Devices.* Institute for Informatics and Telematics of the National Research Council of Italy, Pisa.

Miller, C., Blazakis, D., DaiZovi, D., Esser, S., Iozzo, V., Weinmann, R-P., (2012): *iOS Hacker's Handbook.* Wiley, USA. ISBN: 978-1118204122.

Ramu, S., (2012): *Mobile Malware Evolution, Detection and Defense.* EECE 571B, Term surver paper, April 2012. The Institute for Computing, Information and Cognitive Systems (ICICS), University of British Columbia, Vancouver, Canada.

Rassan, I.A., Al Sheikh, M.A., (2013): *Securing Application in Mobile Computing.* International Journal of Information and Electronics Engineering, Vol. 3, No. 5, September 2013.

Six, J., (2011): *Application Security for the Android Platform.* O'Reilly, USA. ISBN: 978-1-449-31507-8.

Son, Y., Lee, Y., Oh, S., (2012): *Design and Implementation of the Compiler with Secure Coding Rules for Developing Secure Mobile Applications in Memory Usages*. International Journal of Smart Home Vol. 6, No. 4, October, 2012.

Sujithra, M., Padmavathi, G., (2012): *Mobile Device Security: A Survey on Mobile Device Threats, Vulnerabilities and their Defensive Mechanism.* International Journal of Computer Applications (0975 – 8887) Volume 56– No.14, October 2012.

Thurén, T., (2005): *Källkritik*. Liber, Stockholm. ISBN: 978-91-47-05293-6.

Trost, J., (2010): *Kvalitativa intervjuer*. Studentlitteratur, Lund. ISBN: 978-91-44-06216-7.

Wang, Y., Streff, K., Raman, S., (2012): *Security Threats and Analysis of Security Challenges in Smartphones.* Dakota State University, Madison, South Dakota, USA.

Zdziarski, J., (2012): *Hacking and Securing iOS Applications*. O'Reilly, USA. ISBN: 978-1-449-31874-1.