BACHELOROPPGAVE:

MOBILSIKKERHET

FORFATTER(E):

Ernst Kristian Henningsen
Andréas Bålsrud Berg

Dato: 19.05.2010

Sammendrag

Tittel:	Mobilsikkerhet
Dato:	19.05.2010
Forfattere:	Ernst Kristian Henningsen, Andréas Bålsrud Berg
Veiledere:	Jan A. Audestad, Maciej Pietka
Oppdragsgiver:	Sparebanken Hedmark
Kontaktperson:	Ernst Kristian Henningsen
Nøkkelord:	Mobiltelefon, smarttelefon, sikkerhet, Android
Antall sider:	53
Antall vedlegg:	19
Antall sider inkl. vedlegg:	92
Tilgjengelighet:	Åpen

Sammendrag: Fremtiden er mobil. Det er ikke lenger en overdrivelse å si at alle har en mobiltelefon. Snarere tvert imot. Det er i tillegg en stor vekst blant antall personer som blir utsatt for ubehageligheter som digitale tyveri, identitetstyveri og lignende. De fleste har et veldig avslappende forhold til hvordan bruke sin mobiltelefon, og det er derfor dette er en tikkende bombe. Det er bare et spørsmål om tid før kriminelle aktiviteter som før var forbeholdt datamaskinen tar verden med storm også på mobiltelefoner. For å innskrenke konsekvensen av dette bør en derfor ta tak nå, for å gjøre fremtiden innen mobiltelefoni sikrere og behageligere. I denne oppgaven har vi sett på noen av utfordringene som foreligger innen dette.

Rent teknisk har vi fokusert på mer moderne mobiltelefoner, ettersom disse er raskt på vei inn i markedet og har ressurser til å være små datamaskiner. Vi har hatt fokus på operativsystemet Android, men det meste av våre resultater kan allikevel settes opp mot de fleste andre operativsystem innen mobiltelefoni. Prosesserings-potentialet i slike enheter (smarttelefoner) er enorme og dette utgjør i seg selv en stor sikkerhetsrisiko. Det er få begrensninger på hva som er mulig å gjøre på slike system, og de begrensninger som foreligger jobber en med at skal forsvinne. I denne rapporten tar vi for oss svakheter med dette systemet, og hvorfor de er svakheter. Vi mener ikke at teknologiforskningen skal stagnere, men at den skal utvikles i tråd med den økende digitale kriminaliteten. Vi må være bevisste.

Vi vil vise at det i de fleste tilfeller er brukeren av enheten selv som må gjøres bevisst på konsekvensen av sine handlinger. Dette er i seg selv en stor utfordring. Vi diskuterer derfor tilnærminger til hvordan dette kan være mulig å få til på en effektiv og god måte.

Forord

Av de oppgavene som ble presentert for oss som mulige bacheloroppgaver var det denne oppgaven med å finne svake punkt i mobiltelefonens sikkerhet som grep vår interesse fra starten av. Vi tok raskt kontakt med Sparebanken Hedmark, som ble vår oppdragsgiver, for så å videreutvikle oppgavebeskrivelsen til en oppgave vi selv veldig gjerne kunne tenke oss å jobbe med i et halvt år. Vi har hatt en bratt læringskurve og lært mye spennende i løpet av arbeidet med denne oppgava, og håper dette dokumentet kan være lærerikt og samtidig en liten tankevekker for de som leser den.

Vi har ikke stått alene ved utarbeiding av denne rapporten. Vi vil derfor benytte denne mulighet i starten av vår rapport til å utbringe en takk til flere personer vi har hatt en dialog med. Vi synes det passer best med å først trekke frem våre veiledere, Jan Arild Audestad og Maciej Pietka, som begge har pekt oss i riktig retning ved flere anledninger. De har vært svært positive til vår oppgave, noe som har smittet over på oss i form av en god porsjon motivasjon til å stå på. Våre kontaktpersoner hos vår oppdragsgiver Sparebanken Hedmark; Amund Mortensen (Trainee sikkerhet) og Thore Fjogstad (sikkerhetssjef) har støttet oss bra gjennom denne prosjektperioden. De har vært imøtekommende med våre spørsmål når enn det måtte være. Det har vært enkelt å ha en pågående dialog med dem, og dette er noe vi tror har hatt en positiv effekt på vår oppgave. I tillegg fikk vi låne to testtelefoner, Nexus One og HTC HERO som har vært til stor hjelp i testfasen av prosjektet.

Utover dette vil vi takke andre som har hjulpet oss med informasjon til vår oppgave:

- Netcom v/ Daniel Skotheim for informasjon om kontantkortabonnement samt startpakke kontantkort for testing.
- Telenor, Onecall og Chess for informasjon om kontantkortabonnement
- Post og teletilsynet for lov-henvisning og rundskriv rundt kontantkortabonnement
- Slobodan Petrovic (HiG): Undervisningsmateriell i emnet Wireless Communication Security (informasjon om bluetooth)
- Tom Røise (HiG) for hjelp til metodevalg i forhold til systemutvikling ved prosjektet
- Frank Berg og Gunn Marie Henningsen for korrekturlesing av våre dokumenter.

Ernst Kristian Henningsen Andréas Bålsrud Berg

Gjøvik

19.05.2010

Ordforklaringer

- 3G: Tredjegenerasjons mobilnett. Muligheter for rask internettilgang på mobiltelefonen.
- Adware: Program som har innebygget reklame.
- aGPS: Assisted Global Position System. Et system som foruten det vanlige GPS-systemet, også bruker informasjon fra mobile nettverk.
- Android Market: Samlingsplass for utallige applikasjoner for Android. Både gratis og applikasjoner som koster penger. Sistnevnte er ikke tilgjengelig i Norge enda.
- Android SDK: Android Software Developement Kit. Utviklingsverktøy for å lage applikasjoner til Android.
- Autentisering: Bevise at en er den en utgir seg for å være.
- BackTrack: Operativsystem basert på Ubuntu Linux, med mange forhåndsinstallerte programmer som går på sårbarhetstesting, cracking og hacking.
- BankID: En autentisering og signeringsløsning driftet av bankenes betalingssentral.
- Blacklist: Liste over ekskluderte enheter.
- Bluetooth: Grensesnitt for å kommunisere trådløst over korte distanser.
- Botnet: En samling av datamaskiner, infisert av virus/trojanere, som er kontrollert av en tredjepart.
- Bruteforce: Gå gjennom alle mulige kombinasjoner
- Cache: Midlertidig lagringsområde i minnet for data som med stor sannsynlighet vil bli brukt i nærmeste fremtid.
- Daemon: Bakgrunnsprosess.
- Dalvik VM: Dalvik Virtual Machine.
- DDoS: Distributed Denial of Service / Distribuert tjenestenekt angrep. Flere datamaskiner går sammen for å angripe en og samme maskin, og hindre denne i å utføre en gitt oppgave.
- Debugger: Et program som søker etter feil i en gitt programkode.
- Developer-community: Forumer og portaler med utviklere hvor man kan diskutere og dele erfaringer og informasjon.
- DoS: Denial of Service / Tjenestenekt.
- Emulere: Imitere en funksjon, for eksempel et operativsystem. Kjøre dette virtuelt.
- Exploit: Et stykke programkode som utnytter en sårbarhet i et gitt program.
- Glibc: GNU C Library. Bibliotek med programfunksjoner.
- GPL: General Public Lisence (Program som er underlagt denne lisensen må distribueres med åpen kildekode).
- Google checkout: Betalingsløsning fra Google.
- GPRS: General Packet Radio Service. Brukes for å koble en mobiltelefon til internett.

- GPS: Globalt Posisjonerings System. Basert på navigasjonssatellitter for å finne ut hvor man befinner seg.
- GSM: Globalt System for Mobilkommunikasjon. Rundt 82-85% av alle mobiltelefoner benytter GSM.
- GUI: Graphical User Interface.
- Infrarød: Grensesnitt som bruker infrarødt lys for å kommunisere trådløst over svært korte avstander. Brukes sjeldent.
- IP-adresse: Internet Protokoll adresse. Eksempelvis '192.168.1.1'.
- iPhone: Mobiltelefon utviklet av Apple.
- Kildekode: Opprinnelig programkode før den blir kompilert til et kjørbart program.
- LGPL: Lesser General Public Lisence (En versjon av ovenstående lisens som muliggjør bruk uten å oppgi kildekode).
- Linux: operativsystem for PC, på samme måte som Windows.
- Linux-kjerne: "Hjertet" av et Linux-basert operativsystem. Fungerer som hjernen av systemet og kontrollerer hvordan hardware og software kommuniserer.
- MAC-adresse: Media Access Control adresse. En unik adresse for hver nettverksadapter.
- Malware: Malicious Software (skadelig programvare).
- MITM: Man in the middle.
- MMS: Multimedia Messaging Service.
- NMAP (ZENMAP) 5.21: Et program som blir brukt for å søke etter åpne porter.
- NMT: Nordisk mobiltelefonsystem.
- Obfuskere: En handling for å manipulere noe til å bli mindre gjenkjennelig.
- Offentlig IP-adresse: Internet Protokoll adresse utenfor et hjemmenettverk. Det er denne adressen du vil bli identifisert som når man surfer på internett.
- Ormer: Ondsinnet programvare som kan spre seg selv.
- OTA: Over the Air.
- Paring: om Bluetooth; koble sammen to enheter.
- Penetrasjonstest: Gjøre forsøk for å prøve å komme seg inn på et system.
- Pharming: Videresende en bruker fra en gitt nettside til en helt lik men falsk nettside.
- Phishing: forsøke å lure frem sensitiv informasjon fra noen.
- Polymorphisme: Mulighet for å endre seg selv.
- Proxyer: En isolert virtuell tunnel mellom to datamaskiner over et større nettverk.
- Risiko: sannsynlighet for at noe kan skje multiplisert med konsekvensen for hvis det skjer.
- ROM: Et minneområde en ikke kan skrive til, bare lese fra.
- Script: En liste over kommandoer som skal kjøres automatisk.
- Security by obscurity: En handling for å manipulere noe til å bli mindre gjenkjennelig, for at det da skal øke sikkerheten.

- SIM/USIM: (Universal) Suscriber Identity Module. Et lite smartkort som inneholder opplysninger om teleoperatøren og abonnenten. Brukes for å opprette kontakt mellom abonnenten og de forskjellige elementene i mobilsystemet han er knyttet til.
- SMS: Short Messaging Service.
- Sosial manipulering: Manipulere noen enten ved bruk av psykologi eller utgi seg for å være noen andre for å lure noen til å for eksempel oppgi sensitive opplysninger som passord.
- Spam: Søppelpost.
- Spyware: Program som overvåker din datamaskin og sender ut informasjon om deg, for eksempel søkevaner på internett.
- SSP: Secure Simple Pairing
- Stack: en Last In, First Out data struktur.
- Symbian: Operativsystem for mobiltelefoner. Som oftest brukt i Nokia-telefoner.
- SYN-bit: SyncronizeFor å starte en tilkobling til en tjeneste sendes det først en pakke med dette bitet satt.
- Syntaks: Struktur/regelverk for programmeringsspråk.
- Tastelås: Kode for å aksessere selve telefonen uavhengig av SIM-kort.
- TCP: Transmission Control Protocol.
- Telnet: En nettverksprotokoll for å opprette en tekstbasert 2-veis kommunikasjon.
- Trojanere: Enkelte ganger referert til som Trojansk Hest. Malware som ikke kan kopiere seg selv, men ser ut til å ha en ønskelig funksjon for en bruker slik at den blir installert, og da kan fungere som en bakdør slik at andre kan skaffe seg uautorisert tilgang til PC'en/mobiltelefonen.
- Touch-skjerm: En skjerm som reagerer på berøring fra brukeren. Kan benyttes i stedet for eller i tillegg til tastatur.
- UID: User Identification Database.
- USB: Universal Serial Bus. Et grensesnitt for kommunikasjon mellom enheter. Eksempelvis mellom PC og mobiltelefon.
- Virus: Et dataprogram som kan kopiere seg selv og infisere en datamaskin/telefon og gjøre skade.
- Whitelist: Liste over godkjente enheter.
- WIFI: Trådløst nettverk.
- Windows Mobile: Operativsystem for mobiltelefoner, utviklet at Microsoft.

Innhold

1. Innledning

Etter vi gikk inn i informasjonsalderen har produksjonen av teknologiske enheter skutt i været. Datamaskiner spesielt, som før var forbeholdt de få personene som virkelig hadde bruk for det, er nå allemannseie. Det samme gjelder mobiltelefoner. De første automatiske mobiltelefonene i det Nordiske MobilTelefoni-systemet veide omtrent 5 kg. De inneholdt en liten mikroprosessor og hadde en beskjeden prosesserings- og lagringskapasitet. Dette var omkring 1983. Mesteparten av vekten gikk med til batteriet. I dag veier en mobiltelefon noen titals gram og forsvinner nesten i bukselomma. Nye materialer har ført til at batteriene er blitt svært små og kan lagre mye mer energi pr gram batterivekt enn tidligere batterier.

Samtidig er prosesserings- og lagringskapasiteten pr volumenhet blitt enorm på grunn av utviklingen innen halvleder- og mikroprosessorteknologien. Disse mobiltelefonene, bedre kjent som smarttelefoner, kan utføre stort sett de samme oppgavene som en svakere datamaskin kan.

Mobiltelefonen har med årene også utviklet seg til å bli mye mer enn bare en telefon. Tjenester som tidligere bare var forbeholdt datamaskinen som nettsurfing, streaming, e-handel, nettbank etc., blir sakte men sikkert tilgjengelig på mobile enheter som er mye mer portable enn en datamaskin.

Malware og lignende som før har vært mest utbredt på datamaskiner, blir nå mer og mer utbredt på mobiltelefoner. Potensialet for å bli angrepet har så langt vært liten, men det har forekommet noen tilfeller. Noen peker på mangel av standarder og lite ressurser i enhetene som avgjørende for at mobiltelefonen har fått være i fred, men den tiden har forandret seg. Dette er en utvikling som en ikke må se vekk ifra. Det er stadig oftere oppslag i media om virus, trojanere og phishingforsøk som forekommer på mobiltelefoner.

Flere går rett og slett nå om dagen rundt med en liten datamaskin i lomma, uten å være klar over det. Ressurssterke mobiltelefoner, heretter kalt "Smarttelefoner", har kommet for å bli og tar flere og flere markedsandeler. Totalt sett (gjennom smarttelefoner, vanlige telefoner og andre tele-enheter) er det i dag mer enn 4 milliarder abonnenter i GSM/3G[1].

Smarttelefonene sørger for at man alltid er i kontakt med omverdenen. Dette har sine positive og negative sider. De positive er selvfølgelig at det som en før gjorde på datamaskinen, nå kan gjøres mens en venter på bussen på holdeplassen. Denne fleksibiliteten som har blitt implementert i vår hverdag, kombinert med all den personlige informasjonen som flere brukere flittig lagrer på sin smarttelefon, utgjør også den negative siden. Det er flere personer med uærlige hensikter som ser muligheten for å tjene penger på utnyttelse av denne teknologien, for eksempel ved å stjele bankkontodetaljer. Det er derfor veldig viktig å være bevisst på den potensielle risikoen en løper, når en bruker et slikt verktøy.

2. Mål og rammer

2.1. Vår bakgrunn

Gruppemedlemmene er studenter på Bachelor i Informasjonsikkerhet ved Høgskolen i Gjøvik. Vi har ingen tidligere kompetanse på telekommunikasjon eller mobiltelefoner, men valgte denne oppgaven fordi vi så på det som en fin utfordring hvor vi kan lære mye nyttig som kan anvendes senere i arbeidslivet. Oppgaven var en av flere forslag som ble foreslått til Høgskolen i Gjøvik. Vi har begge studert emnene "Ethical Hacking and Penetration Testing" og "Information Warfare" som vi tenker gir oss bedre kompetanse for dette prosjektet.

2.2. Vår oppdragsgiver

Sparebanken Hedmark har vært vår oppdragsgiver for denne Bacheloroppgaven. De er den ledende leverandøren av finansielle tjenester til personer, bedrifter og offentlig sektor i Hedmark. De har 500 ansatte som fordeler seg på 27 kontorer rundt om i Hedmark.

I tillegg til bank-, plassering- og forsikringstjenester tilbyr de tjenester innen eiendomsmegling, leasing og annen finansiering.

Mobiltelefon er et stort satsningsområde, der de tilbyr blant annet SMS-bank og Mobilbank.

Våre kontaktpersoner i banken har vært stasjonert i Hamar, som også er hvor konsernadministrasjonen ligger.

Sparebanken Hedmark er medeier i SpareBank 1 Gruppen AS.

2.3. Øvrige roller

Oppdragsgiver:

- Sparebanken Hedmark:
  - Amund Mortensen
  - Thore Fjogstad

Veiledere:

- Jan Arild Audestad
- Maciej Pietka

Gruppemedlemmer:

- Ernst Kristian Henningsen - Gruppeleder
- Andreas B. Berg – Referent

2.4. Prosjektmål

2.4.1. Effektmål:

Øke Sparebanken Hedmarks kompetanse og kunnskap om mobiltelefonsikkerhet, samt gjøre ansatte og kunder mer bevisste om risikoen ved bruk av mobiltelefoner som verktøy.

2.4.2. Resultatmål:

Produsere en rapport som gjør det enkelt å sette seg inn i de forskjellige sikkerhetsmekanismene og risikoene ved bruk av smarttelefoner, med spesiell vinkling mot operativsystemet Android.

Produsere materiell som har som formål å bevisstgjøre Sparebanken Hedmarks kunder på mobiltelefonsikkerhet.

2.5. Rammer

Testfase som inneholder eksperimentering med smarttelefoner, hvor det er sjanse for spredning av skadelig kode eller lignende, må gjøres i et isolert miljø.

Prosjektet skal være ferdig 20.05.10

3. Omfang

3.1. Oppgavebeskrivelse

Etter oppdrag fra Sparebanken Hedmark, skal vi ta for oss mobilsikkerhet med fokus på smarttelefonen, og undersøke hvilken risiko som foreligger ved bruk, spesielt i forhold til operativsystemet Android. Dette som et ledd av deres økende satsning på mobiltjenester. Vi vil dykke ned og se på hvilke relevante sikkerhetsmekanismer som finnes i dette operativsystemet, og om disse er så sikre som en skal ha det til. Vi vil blant annet se på muligheter for angrep gjennom SMS, MMS, Bluetooth og WIFI, i tillegg til å relatere disse angrepene til det aktuelle operativsystemet.

Ut ifra dette vil vi utvikle en detaljert rapport, samt annet relevant materiell. Dette vil kunne brukes av Sparebanken Hedmark for å forbedre deres kompetanse på området og skape fokus på mobilsikkerhet. Vi vil foreta sårbarhetstesting over WIFI samt forsøke å utvikle en ”proof-of-concept”-applikasjon som viser hva som kan være mulig hvis en får skadelig applikasjon installert på telefonen.

Mot slutten av prosjektet vil vi utarbeide en presentasjon der vi vil demonstrere mulighetene en angriper har hvis han/hun oppnår tilgang til systemet, samt hvordan dette eventuelt er mulig.

Våre problemstillinger er som følger:

- Hvilke muligheter finnes for å kompromittere/angripe en mobiltelefon gjennom SMS, MMS, Bluetooth og WIFI relatert til operativsystemet Android?
- Hvilke tiltak kan innføres for å sikre disse og forhindre uautorisert tilgang og sikkerhetsbrudd på telefonen?

3.2. Avgrensning

3.2.1. Operativsystem

Ettersom operativsystemet står sentralt i smarttelefonene, diskuterte vi på vårt første møte med oppdragsgivere og veiledere, hvilke operativsystemer som kunne være aktuelle. Vi kom frem til at det stod mellom Windows Mobile, Android, Symbian og iPhone. Vi ble på dette møte også enige om at vi burde ta for oss et enkelt operativsystem, ettersom vi har begrenset med tid.
Sparebanken Hedmark stilte seg åpen for alle operativsystemene, det ble derfor opp til gruppa å velge.
Vi satte opp flere krav som operativsystemet måtte tilfredsstille. Kravene var som følger:

- Operativsystemet bør ha stor markedsandel/brukermasse, eller ha potensial til å oppnå dette innen et par år.

- Operativsystemet bør være lett tilgjengelig.

- Operativsystemet bør ha mulighet for å kjøre egenproduserte applikasjoner.

- Operativsystemet bør ha åpen kildekode, slik at sikkerhetsmekanismer kan studeres på lavere nivå hvis nødvendig.

Basert på dette kunne vi raskt luke vekk både Windows Mobile og iPhone, ettersom deres kildekode er lukket. Symbian sin kildekode var lukket helt frem til februar 2010 da Nokia tok over med sin visjon om å gjøre operativsystemet til åpen kildekode[2]. Vi sto derfor igjen med Symbian og Android (som også har åpen kildekode).

En stor forskjell på Symbian og Android er at Symbian allerede har en stor markedsandel, mens Android ikke har dette per i dag (januar 2010). Ettersom vi har tro på at Android vil ta

store markedsandeler de kommende årene endte vi med å velge dette operativsystemet. Det var også avgjørende for oss at det ikke har blitt gjort grundig forskning på Android, ettersom det ikke har vært tilgjengelig mer enn noen få år.

3.2.2. Grensesnitt

Mobildatakommunikasjon

Vi har i denne oppgaven valgt å ikke rette vår oppmerksomhet mot selve mobildatakommuniksjonen, med unntak av SMS og MMS. Vi har blant annet lest gjennom kompendiet til vår veileder, "Network Security, Lecture Notes, Gjøvik University College, 2010, Jan A. Audestad" og har fått inntrykk av at det på dette området er tilsynelatende god sikkerhet. Vi må derimot opplyse om at vårt inntrykk muligens ikke lenger vil være gjeldene ettersom det for tiden (06.05.2010) går rykter i media angående at GSM-koden er knekt[3].

Fysiske grensesnitt

Vi har valgt å rette dette prosjektet mer mot fjern-tilkobling. Vi vil derfor nedprioritere å se på muligheter ved direkte fysisk tilgang til en mobiltelefon, som for eksempel gjennom USB, minnekort, SIM/USIM, infrarød og tastatur/touch-skjerm.

4. Operativsystem

4.1. Android

Figur 4.1.1

Operativsystemet Android (logo - figur 4.1.1), utviklet av Google, er basert på en linux-kjerne. Operativsystemet er på tross av dette ikke linux. Android bruker ikke linux sitt bibliotek, men et som er laget av Google selv.[4] Dette biblioteket, "Bionic", har lukket kildekode i motsetning til linux sitt bibliotek som er åpent. Google mener deler av grunnen for dette valget er fordi biblioteket i linux (glibc) er basert på GPL-lisensen (General Public Lisence) som sier at applikasjoner som bruker åpen kildekode, også må gjøres åpen for allmennheten. Dette er det flere bedrifter som ikke setter pris på. De vil ikke vise alle hvilken kode som ligger bak deres applikasjoner. Derfor har Android gitt utviklere mulighet til å tjene penger på applikasjoner solgt til Android, uten å måtte gi opp deres kildekode. Derimot er ikke biblioteket glibc underlagt GPL men LGPL[5] (Lesser General Public Lisence), dette betyr egentlig at en har lov til å bruke glibc til tross for at en selv bruker lukket kode. Dette argumentet i seg selv synes vi derfor ikke holder. De påpeker derimot også på andre grunner som vi synes er høyst relevante, nemlig at glibc blant annet er et stort bibliotek. Antakeligvis for stort sett i forhold til at den skal brukes i små enheter med begrensede ressurser. Google hadde derfor behov for et bibliotek som var optimalisert for slike enheter.

Det er mulig å lage egne applikasjoner til Android. Google oppfordrer til dette gjennom deres developer-community[6]. Her gis det instruksjoner på "hvordan komme i gang".

Applikasjoner kan lastes ned fra Android Market. Disse er signert av utvikleren selv for å kunne opprette tillitsforhold mellom applikasjoner. To applikasjoner kan stole på hverandre ettersom de er utviklet av samme person/firma.

Disse applikasjonene (som på datamaskinen) utgjør i utgangspunktet en stor sikkerhetsrisiko, derfor har Android flere implementerte sikkerhetsmekanismer for å hindre ugagn.

Android praktiserer godkjenning av ressurser. Det å aksessere for eksempel kontaktlista eller sende SMS blir regnet som å aksessere en begrenset ressurs. [7] For at en applikasjon skal kunne bruke en begrenset ressurs må dette deklareres i ei manifest fil.(AndroidManifest.xml). Applikasjonen får ikke tilgang til ressurser som ikke er deklarert i manifest-fila. Ved installasjon av en ny applikasjon, vil brukeren av mobilen bli forespurt om han/hun godtar at de forskjellige ressursene blir tilgjengelig for applikasjonen. En kan ikke delvis godta ressursene. Det vil si, en må enten godta at applikasjonen kan bruke alle ressursene som forespurt, eller så får en ikke installert applikasjonen.

Når installasjonen er gjort kan ikke ressurstilgangen endres uten å installere applikasjonen på ny.

Hver applikasjon blir tildelt en UID (user ID).

To applikasjoner kan ikke ha samme UID, men de kan bli signert med sharedID. Dette innebærer at flere applikasjoner kan samarbeide. I et samarbeid kan begge applikasjonene aksessere ressursene til den andre applikasjon. Dette medfører en potensiell sikkerhetsrisiko, ettersom en ikke blir videre fortalt at to applikasjoner samarbeider ved installasjon[7]. En av applikasjonene kan for eksempel forespørre tilgang til internettressursen. Den andre applikasjonen kan forespørre tilgang til kontaktlista. Hver for seg er sjansen stor for at installasjon blir akseptert. En tenker seg derimot om et par ganger dersom en applikasjon både vil ha tilgang til internett og kontaktlista.

Vi må derimot også legge til at vi ikke har studert koden bak de overnevnte sikkerhetsmekanismene. Informasjonen rundt dette er hentet fra offisielle Android-kilder.

4.2. Oppgradering av Android

Når man kjøper en ny telefon med Android kan man bli sittende med alle mulige versjoner av operativsystemet, alt fra v1.5 til v2.1. De forskjellige telefonprodusentene, i hvert fall HTC, Samsung og Sony Ericsson har sine egne GUI[A] sydd inn i Android. Når Android slippes i nyere versjon tar det ofte tid å få sydd GUI inn i den nye versjonen og man blir da liggende etter med et eldre operativsystem. Eksempelvis har den ene av våre testtelefoner (HTC HERO) per i dag (mai 2010) ikke fått noen oppdatering fra HTC på 7 måneder, og kjører fortsatt v1.5. Det skal vissnok komme en oppgradering til 2.1 i løpet av juni/juli 2010, men dette gjenstår å se. Sony Ericsson X10 ble lansert med v1.6, etter at Android 2.1 var tilgjengelig, og det er her snakk om en oppgradering til v2.1 først til høsten. Altså over et halvt år etter Android 2.1 ble tilgjengelig fra Google[8]. Når det gjelder Nexus One, som var den andre testtelefonen vår, ble denne lansert med det nyeste operativsystemet (v2.1 per april 2010). Vi opplevde å få en oppdatering til den med en gang vi skrudde den på og koblet den til nettverk, dette var trolig mulig fordi Nexus One ikke har noe videreutviklet GUI slik som for eksempel HTC HERO har, og bruker den som kommer som standard i Android. Derfor må ikke produsentene gjøre noe videre tilrettelegging etter en oppdatering har blitt gjort tilgjengelig. Her mottas det altså sikkerhetsoppdateringer rett etter de lanseres, noe som vi anser som bra. Når det gjelder oppgraderinger til nyere kommende operativsystemer, er vi ikke sikre på hvordan dette vil foregå. Vi vil anta at dette vil kunne bli tilgjengelig raskt etter lansering siden det ikke må tilrettelegges noen ny GUI.

At det for de fleste telefonprodusentene tar lang tid å komme med en oppgradering, synes vi er en potensiell sikkerhetsrisiko, spesielt hvis det blir oppdaget sikkerhetshull og sårbarheter. Hvor lang tid det ville tatt før disse ble tettet/fikset, kan en bare spekulere i. (Da med tanke på de produsenter som legger inn eget GUI i sine telefoner).

4.3. Rooting av Android-telefoner – fordeler og ulemper

4.3.1. Hva er rooting?

Rooting av telefonen er en måte å hacke telefonen slik at en vil kunne kjøre applikasjoner og script som administrator. Man vil kunne få skriverettighet til filene i operativsystemet som man normalt ikke har tilgang til, på både godt og ondt[9].

Når det er snakk om rooting på Android må vi ha i tankene at Android ikke er vanlig Linux. Måten det fungerer med root-tilgang på Android er gjennom en tilleggsapplikasjon. Programmer som trenger root-tilgang spør da dette programmet om lov, og brukeren må selv akseptere hvert enkelt program, eller legge de i en ”godkjent-liste”. Det er da altså ikke snakk om noe root-passord som på Linux og iPhone. Man er her sikret med at brukeren må

[A] Graphical User Interface

fysisk trykke på skjermen for å tillate programmene tilgang. Dette programmet går under navnet ”Superuser” og kjører skriptene som ligger under /sbin/.

4.3.2. Fordeler med root-tilgang:

- Mulighet for å flytte cache og programmer til SD-kortet. Gir bedre plass på telefonen for applikasjoner fra Android Market og lignende.
- Legge til nye temaer på telefonen
- Muligheten for å slette applikasjoner på telefonen som var installert ved kjøp av telefonen.
- Oppgavebehandler for root. Gir deg muligheten til å stoppe hvilken som helst bakgrunnsprosess.
- Kan emulere en terminal med root-tilgang på telefonen.
- Ta sikkerhetskopi av hele telefonen

4.3.3. Ulemper med root-tilgang:

- Hvis du ikke forstår hva du gjør, kan du være uheldig nok til å gjøre om telefonen din til en dyr papirvekt.
- Har lommeboken din råd til å erstatte telefonen hvis du er uheldig (punktet ovenfor)?
- Gjøre kjernefiler korrupte
- Stoppe OTA oppdateringer fra å komme til telefonen
- Gjøre telefonen mer sårbar for malware (hvis man ikke bruker noe superuser permission program)
- Rooting kan gjøre garantien på telefonen ugyldig.

Vi testet selv med å roote vår testtelefon HTC Hero. Dette gikk helt smertefritt, men det ble en del lesing av forskjellige forumer[10, 11] for å være helt sikre på at vi ikke gjorde noe feil. Vi fikk testet i praksis hvordan Superuser fungerer. Vi fant ut at dette er veldig enkelt i bruk. Trenger en applikasjon root-tilgang spretter det opp et nytt vindu som spør om du vil tillate dette programmet å kjøre som root.

Rooting av en telefon burde ikke gjøres hvis man ikke har kompetanse og forståelse for hva dette vil innebære. Man vil selvfølgelig gjøre telefonen mer sårbar hvis man ikke tar nødvendige sikkerhets forhåndsregler som å ha et program som styrer root-tilgang for programmer, og ikke installere programmer som vil ha tilgang til ressurser de ikke ”trenger” (kan være malware). Vi vil ikke dele fremgangsmåten for å roote en telefon her i dette dokumentet, men hvis man er interessert i dette så har Google svarene. Rooting av telefoner gjøres altså på eget ansvar.

4.4. Dalvik VM

Når en applikasjon blir startet, kjøres den i Dalvik Virtual Machine (Dalvik VM). Dette er en registerbasert virtuell maskin designet og lagd av Dan Bornstein[B]. Det tillater raskere eksekvering av mindre programmer. Siden Dalvik VM lagrer til registeret i stedet for stack'en håper Google på at det vil bli utført 30 prosent færre instruksjoner ved kjøring, og at det derfor vil fungere raskere[12].
Dalvik er optimalisert for å kjøre på enheter med lite minne og å kjøre mange instanser av virtuelle maskiner samtidig[13]. Dalvik kjører Java plattformen på Android, hvor alle applikasjonene på systemet som har blitt konvertert til et kompakt Dalvik Executable (.dex) format kjøres. Den opprinnelige koden Dalvik's kjerne-biblioteker er skrevet i C/C++ og fungerer i et Linux-miljø uten modifikasjoner.

Når hver applikasjon kjøres i sin egen virtuelle maskin, oppnås en god sikkerhetsgevinst. De forskjellige applikasjonene kan sies å kjøre i et sandbox-miljø. Dette innebærer at en applikasjon ikke har tilgang til noe annet enn seg selv og de ressursene som ble akseptert ved installasjon. Hver applikasjon vil også ha sitt eget adresseområde i minnet og kjøre som en egen prosess. Fordelen med dette er hvis et program skulle inneholde en svakhet som muliggjør eksekvering av vilkårlig kode (noe som i så fall er en stor sikkerhetsrisiko) vil angriperen ikke få tilgang til noe annet enn det selve applikasjonen har tilgang til.

4.5. Android Market

Android Market er det offisielle området hvor en kan laste ned applikasjoner til sin Android-telefon, både betalbare og kostnadsfrie applikasjoner. (Betalbare applikasjoner er ikke tilgjengelige i Norge per 26.02.2010).

Vi tror at de fleste som vil ta i bruk nye applikasjoner bruker Android Market, derfor er det viktig at applikasjonene som er mulig å laste ned er legitime og ikke av ond art. Hvordan er dette derimot mulig å overholde?

Et tilfelle hvor en ondartet applikasjon fikk mulig til å bre seg ut gjennom Android Market var i desember 2009. En bruker som kaller seg Droid09, laget en applikasjon som hadde som formål å samle inn bankkonto detaljer[14]. Applikasjonen ble kort tid etter tatt vekk fra Android Market og brukeren ble ekskludert. Det er ikke kjent om noen ble offer for phishingforsøket. En applikasjon som bare er åpen for nedlasting gjennom Android Market i et par dager, for ikke å si timer, kan skape store tap blant Android-brukerne og for Google selv som kan miste markedsandeler. Utfordringen er derfor å hindre at slike applikasjoner blir publisert på Android Market i første omgang, altså en proaktiv mekanisme.

[B] http://www.milk.com/home/danfuzz/resume/

Følgende prosesser/mekanismer har Google tatt i bruk for å hindre publisering av malware:

- For i det hele tatt å kunne laste opp på Android Market må en ha en registrert bruker.
- For å kunne registreres må en gjennom følgende prosess (per 12.02.10):
- En må logge inn med en gyldig Gmail-konto (din egen eller en annen sin..)
- En må registrere en utvikler-profil med følgende informasjon
- Utviklerens navn (vil bli distribuert med opplastet applikasjon)
- Epost-adresse
- Hjemmeside
- Telefonnummer (hvis det skulle oppstå problem med opplastet applikasjon)

En må betale en registreringsavgift på 25dollar gjennom Google sin egen betalingstjeneste, Google Checkout. En må her oppgi følgende[15]:

- Hvilket land
- Kortnummer
- Utløpsdato
- CVC-nummer
- Navn på korteier
- Faktureringsadresse
- Postnummer
- Poststed
- Telefonnummer (brukes for å bekrefte konto)
- Velge leveringsadresse (faktureringsadresse eller annen)
- Godta vilkårene som blant annet tilsier at du må være over 18 år.

Med all denne informasjonen er det i teorien mulig å koble en applikasjon entydig til en person. For en kriminell som ikke vil bli tatt er dette dårlige nyheter. I praksis er det derimot en helt annen sak.

Første steg med å logge inn med en Gmail-konto kan enkelt gås rundt ved å lage en Gmail-konto der og da med en fiktiv bruker eller bruke en annens konto (som en har skaffet seg innloggingsdetaljert til ved for eksempel keylogging eller social engineering).

Neste steg er å betale 25dollar. Dette er en sum som de fleste ikke vil ha noe imot/ha noen vanskeligheter med å betale. Derimot vil betaling med kort også kunne spores entydig til en person. En kriminell kan derfor bruke en annens kort-detaljer, som for eksempel kan kjøpes på det svarte markedet[16]. Telefonnummeret som blir brukt ved kontobekreftelse kan også være en annens, ved at en rett og slett har stjålet en telefon. For å luke vekk de enkle tilfellene av svindling, regner vi med at Google foretar kryssjekking av informasjon oppgitt.

En svindler må derfor koordinere brukerinformasjon han oppgir noe som er mulig for den målrettede.

Når en har en registrert bruker kan det lastes opp applikasjoner. Ettersom prioriteringen bør være å hindre at malware blir publisert gjennom Android Market, er det mekanismene etter selve registreringen som bør være sikre. En kriminell kan jo registrere all informasjon på seg selv, for så å «gå i skjul» etter applikasjon er opplastet, eventuelt påstå at noen andre har benyttet seg av hans/hennes identitet. En ville i sistnevnte tilfelle gjerne gått gjennom flere proxyer for å obfuskere[C] hvor egentlig registreringen ble foretatt.

Det vi regner som det sikreste, hadde vært om Google brukte implementerte mekanismer som sjekker applikasjoner som blir publisert. Den eneste mekanismen Google har ved publikasjon av applikasjon er om applikasjonen er teknisk i orden. Altså om den vil kunne kjøre på en Android telefon, en slags debugger med andre ord. Usikker kilde på nettet[17] sier at Google ikke har noen mekanismer som omhandler dette og at de har lagt mye av dette ansvaret over på brukerne. Brukerne må altså ta kontakt hvis en applikasjon ikke fungerer som den skal. Dette er noe som vi anser som et hull i Android's sikkerhetsmekanismer. Det må derimot også informeres om at slike proaktive mekanismer, som sjekker en applikasjon for ondsinnet kode, kan være svært vanskelige å implementere. Dette fordi mekanismen ikke bare må sjekke linje for linje for ondsinnet kode, men også må sjekke linjer i forhold til hverandre. Vi forestiller oss om det vil komme en bølge av ondsinnede applikasjon publisert på Android Market før Google implementerer en slik mekanisme.

Det er nå snakk om Android v2.2 og endringer i Android Market. Det er fortsatt usikkert hvordan disse endringene blir, men det er snakk om muligheter for aktivering av automatiske oppdateringer av programvare. Hvis dette blir lansert, så blir det selvfølgelig enklere å bruke programmer, siden man slipper å gå manuelt inn og oppdatere hvert eneste program. Dette vil derimot også skape en sikkerhetsrisiko. Man kan laste ned og installere en uskyldig liten applikasjon og ta den i bruk med de ressursene den trenger for å kjøre, men utvikleren kan senere legge inn krav om flere ressurser, eller kode om programmet til å gjøre noe helt annet. Dette vil bli automatisk oppdatert hos brukeren uten at brukeren er klar over hvilke endringer som kan ha blitt gjort. Android 2.2 er ikke lansert som kode enda (april 2010), men funksjonen er beskrevet av Android testere[18].

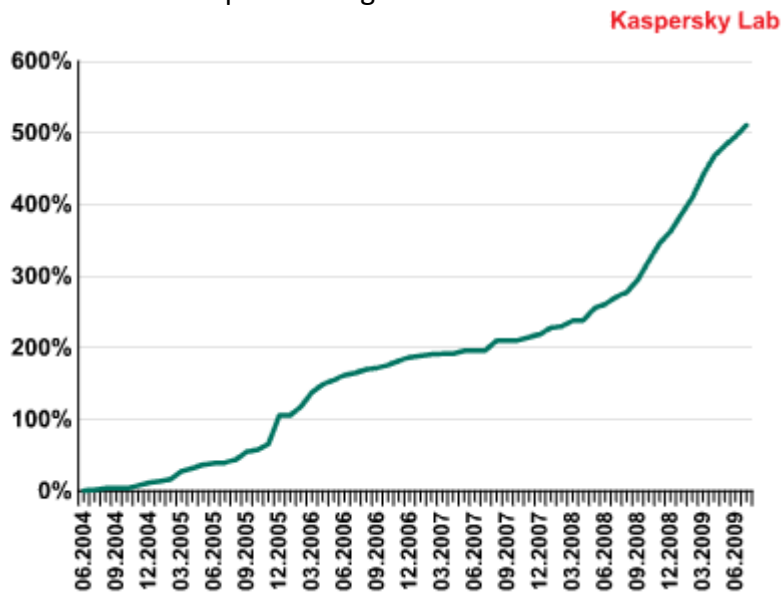
5. Angrep

5.1. Malware

Ordet malware er en sammensetning av ordene malicious (skadelig) og software (programvare), noe som altså sier at malware er programvare med onde hensikter.
De første malwaren til smarttelefoner dukket opp i juni 2004, og i tiden herfra og til 2006 vokste antallet betydelig. Bare de siste tre årene har antall forskjellige familier/typer

[C] En handling for å manipulere noe til å bli mindre gjenkjennelig

malware for mobiltelefoner hatt en vekst på 202%[19] (figur 5.1.1). Det er for tiden Symbian som innehar størsteparten av mobilmarkedet (Figur 5.1.2), og det er da også på den platformen det finnes mest malware. Men med tanke på Android's potensielle vekst i årene fremover, tror vi antallet vil vokse pent her også.



Figur 5.1.1

Vekst i kjente malware-modifikasjoner (2004-2009)

Platform	Number of families	Number of modifications
Symbian	62	253
J2ME	31	182
WinCE	5	26
Python	3	45
SGold	3	4
MSIL	2	4

Figur 5.1.2

(hentet fra viruslist.com. Kaspersky Lab 2009)

Families: familier/virustyper

Modifications: forskjellige variasjoner av en virusfamilie

Det finnes mange forskjellige typer malware som: virus, ormer, trojanere, spyware, adware, bakdører, spam, phishing, pharming, keyloggere og DDoS[D20].

I følge Android Security Team[E] er malware en applikasjon som:

- Tømmer batteriet veldig fort
- Viser brukeren uoppfordrede meldinger (spesielt meldinger som oppfordrer brukeren til å kjøpe noe)
- Motstår (eller forsøker å motstå) brukerens forsøk på å avinstallere det
- Forsøker automatisk å spre seg selv over til andre enheter
- Gjemmer sine filer og/eller prosesser
- Utgir sensitive opplysninger til en tredjepart, uten at brukeren vet det eller gir sitt samtykke
- Ødelegger brukerens data (eller telefonen selv) uten at brukeren vet det eller gir sitt samtykke
- Utgir seg for å være brukeren (som å sende meldinger/epost eller kjøpe ting fra en nettside) uten om at brukeren vet om det eller gir sitt samtykke
- Ellers på noen annen måte ødelegger brukeropplevelsen med enheten.

I følge Kaspersky Lab[19] så kunne malware for 3 år siden:

- Spres via Bluetooth, MMS
- Sende SMS meldinger
- Infisere filer
- Aktivere fjernstyring av smarttelefonen
- Modifisere eller erstatte ikoner eller systemapplikasjoner
- Installere ”falske” eller ikke-fungerende fonter og applikasjoner
- Bekjempe antivirus-programmer
- Installere annen skadelig programvare
- Låse minnekort
- Stjele data

Malware for mobiltelefoner har nå adoptert nye teknologier og teknikker de siste tre årene, og inneholder nå blant annet følgende:

- Spredning via flyttbare medier (minnekort/minnepinner)
- Ødelegge brukerdata
- Deaktivere sikkerhetsmekanismer i operativsystemet
- Laste ned andre filer fra nettet
- Ringe betalingstjenester
- Polymorphisme[F]

[D] Distributed Denial of Service (tjenestenekt)
[E] http://groups.google.com/group/android-security-announce
[F] Mulighet for å forandre seg selv

5.2. SMS

Short Messaging Service er noe som har eksistert en god stund og har vært flittig brukt av de fleste. I starten kunne en SMS bare inneholde 160 tegn, slik at en måtte skrive en ny SMS hvis innholdet overskredet dette. Teknologien har derimot gjort fremskritt og det er nå mulig å emulere SMS-er som inneholder flere tegn. En trenger altså ikke skrive to SMS-er hvis innhold tilsvarer for eksempel 210 tegn. Derimot må en betale for sending av to SMS-er, ettersom det faktisk blir sendt to, forskjellen er at de er "koblet sammen".

Ettersom SMS er så utbredt som det er, kunne et vellykket angrep gjennom SMS fått store konsekvenser. Heldigvis har vi ikke sett mye til direkte angrep gjennom SMS. Derimot har det blitt mer og mer utbredt å bruke SMS som en kanal for blant annet spredning av Malware. En mottar altså da en SMS med en skadelig link. Går en inn på gjeldende adresse laster en for eksempel ned en skadelig applikasjon. Vi har ikke funnet noen tilfeller der en ved bruk av for eksempel exploits, kan manipulere en mottaker uten at selve brukeren av telefonen må gjøre noe aktivt. Dessverre har metodene/teknikkene for å distribuere farlig kode blitt mer sofistikerte og moderne. Nå er det brukeren som står i fokus, og derav hvordan lure brukeren til å aksessere en link som blir sendt på SMS, for så å trykke aksept for nedlasting/innstallasjon av applikasjoner. Det er altså ikke lenger teknologien som må sette en stopper for skadelige applikasjoner, det er brukeren selv. Dette innebærer en stor sikkerhetsrisiko ettersom det er en mangfoldig brukermasse der ute som ikke har inngående kjennskap til mobil/smarttelefon-sikkerhet. Når dette er sagt, betyr ikke dette at en person som kan anses som oppegående innen informasjonssikkerhet, ikke kan bli lurt av slike phishing-metoder. Phishing handler om å være troverdig. Er det troverdig nok, biter alle på.

Ta siden http://luresms.com/ som eksempel. Denne siden lar hvem som helst sende en anonym SMS til hvem de vil, en kan også velge hvem avsenderen skal være. "Banken", "Politiet", "Skattedirektoratet", etc. er flere eksempler på avsendere som kunne blitt brukt, og derav betraktelig økt sjansen for et phishing-angrep.

Eksempel:
"

Avsender: Skattedirektoratet

Melding: Dette er en automatisk generert melding. I forbindelse med årets selvangivelse kan vi ikke se å ha mottatt restskatt fra deg. Vi må be deg om omgående å betale dette inn som angitt i heftet om selvangivelse du har mottatt fra oss. Skulle det være tilfelle at du ikke har skatt utestående til betaling har det skjedd en glipp i vår oppdatering. For at dette da ikke skal bli sendt automatisk videre til namsmannen ber vi deg svare på denne melding med din personlige kode som du bruker på minside.no i tillegg til pinkode 16 fra skattekort datert 12.12.2008. Dette for å identifisere deg entydig. Vi gjør deg oppmerksom på at

meldingen er gratis hvis du svarer fra mobiltelefon registrert på minside.no (Telefonnummer: 987 654 32). Mvh Skattedirektoratet.

”

Ovenstående er en melding komponert av oss.

5.3. MMS

Det ser ikke ut til at MMS (Multimedia Messaging Service) har vært en populær måte å spre virus på. Det har vært noen episoder gjennom tidene med spesielt henvisning til 2005 da CommWarrior.a-viruset brøt ut.[21] Dette skal være det første viruset som ble spredt gjennom MMS og ble derfor møtt med stor skeptisisme siden dette ikke før hadde inntruffet. Viruset var laget for å kjøre på Nokia’s operativsystem Symbian.

En sto ovenfor en potensiell pandemi på mobil-fronten ettersom viruset i teorien kunne smitte minst like raskt som over epost, hvis ikke raskere siden en sjekker meldinger mottatt på telefon oftere en epost. Før dette hadde mobilvirus bare blitt sett spredt gjennom Blueooth, der det var begrenset hvor langt og hvor fort viruset kunne spre seg. En pandemi inntreffet derimot ikke og vi har ikke sett stort til MMS-virus i ettertid med unntak av avanserte versjoner av dette CommWarrior-viruset. En grunn for dette kan være at dagens telefoner opererer med vidt forskjellige operativsystem. Skal en oppnå stor spredning av et virus må en derfor lage en hybrid som sjekker hvilket operativsystem det er og kjører aktuell kode deretter. I tillegg håper vi at dagens brukere har fått en mer bevisst holding til sikkerhet, men det er ikke noe vi kan si med sikkerhet.

Det er viktig å presisere at selve spredningen av viruset tok sted ved at en mottok en MMS melding der en ble forespurt om å installere gjeldene applikasjon. (en .sis fil). Brukeren selv kunne derfor velge å *ikke* installere viruset. Viruset var derimot skjult bak tilsynelatende ”virus-frie” applikasjoner, så brukerne ble derfor lurt til å installere. Vi har ikke funnet noen indikasjoner på at det er mulig å infisere eller utnytte en telefon gjennom MMS slik at gjeldene telefon blir smittet uten brukerdialog. Vi mener derfor at også her vil det være brukeren som må opptre forsvarlig og ikke installere applikasjoner som kan inneholde skadelig kode. Spredningen av virus har blitt mer moderne og sofistikerte og har derfor gått fra å være sterke kodemessig, til å være sterke med tanke på troverdighet. Det enkleste for en virusprogrammerer er hvis et offer laster ned sin applikasjon fra vedkommendes hjemmeside for så å installere virusapplikasjonen selv. Det er nærmest slik det skjer i dag, bare at det skjer gjennom andre medier som for eksempel Android Market og iPhone sitt marked ”App Store”.[22]

5.4. WIFI

Vi nevnte i starten av denne rapporten at mobiltelefonen er som en liten håndholdt datamaskin. De senere årene har det blitt standard å levere disse telefonene med WIFI. Dette ser vi på som noe positivt ettersom en da kan "surfe gratis" på internett. Hvis man har trådløst nettverk hjemme kan man enkelt og greit koble til sin mobiltelefon på samme måte som med laptopen. Dette gjelder følgelig også ved arbeidsplassen. De samme angrepene som er mulig mot en laptop som bruker WIFI er mer eller mindre aktuelle også mot en mobiltelefon.

Under vil vi trekke frem potensielle angrep som kan bli foretatt mot en mobiltelefon.

Sniffing

Et angrep som går ut på å overvåke et eller flere WIFI-nettverk for personlig informasjon. Dette er et angrep som kan gjøres i passiv modus. Det vil si at en bare mottar datapakker og ikke sender som innebærer at en ikke kan bli oppdaget.

Man in the middle (MITM)

Et angrep som går ut på å stille seg mellom to parter i et WIFI-nettverk. Oftest referert til som part A: Alice og part B: BOB. En tredje part sender dermed datapakker til Alice der en utgjør seg for å være Bob, og sender i tillegg datapakker til Bob der en utgjør seg for å være Alice. Resultatet er at all informasjon går gjennom en tredjepart. Denne tredjeparten bestemmer hva de andre to partene skal se og få av informasjon.

Warsucking

Et angrep som går ut på at en utgir seg for å være et legitimt tilkoblingspunkt for internett. For eksempel på et hotell. En har i dette tilfellet satt opp sitt eget nettverk med internett-tilknytning som tilbyr klienter som kobler seg til, tilgang til internettet. Dette er i prinsippet et angrep som vi anser som å være "i gråsonen" med tanke på ulovlige aktiviteter. Dette fordi en enkelt og greit tilbyr andre å bruke ens eget internett, som ikke er ulovlig i seg selv. Sikkerhetsrisikoen derimot, er som ved MITM. En kan se det klientene foretar seg på internettet og manipulere dette.

Denial of Service (DoS) / Tjenestenekt-angrep

Et angrep som går ut på å forespørre en enhet om informasjon, ofte på en slik måte at enheten går tom for ressurser. Dette resulterer i at enheten responderer svært tregt og i verste fall ikke klarer kommunisere over dette grensesnittet. Konsekvensen av dette er derimot ikke direkte farlig, med mindre det er et organisert angrep. Hvis for eksempel alle

kommunikasjonskanaler skulle bli slått ut under et terrorist-angrep, ville ikke folket ha mulighet til å få raske oppdateringer på nåværende status. Panikk vil da oppstå. Der de andre nevnte angrep er rettet mot konfidensialitet og integritet, er dette angrepet rettet mot tilgjengelighet.

Fordelen med mobiltelefoner med tanke på WIFI-sikkerhet er at de sjelden innehar server-tjenester. Dette betyr at det er færre muligheter for å angripe en slik enhet. En laptop med for eksempel Windows, kjører flere server-tjenester som standard. Tjenester som for eksempel skal gjøre opplevelsen av et datamaskin-nettverk mer brukervennlig. Dette kan i noen tilfeller utnyttes eller "exploites" som også er et ofte bruk fagord.

Ulempen med mobiltelefoner, er at de sjelden blir skrudd av. De vil derfor i teorien være åpen for angrep nærmest 24timer i døgnet – hvis de er tilkoblet et WIFI-nettverket vel og merke.

5.5. Bluetooth

Utviklingen av Bluetooth-teknologien startet først i Skandinavia. Navnet kommer av den danske kongen Harald Bluetooth som spillet en viktig rolle i gjenforeningen av det Skandinaviske Europa[23]. De fleste bærbare datamaskiner og mobiltelefoner blir gitt ut med Bluetooth i dag. På grunn av dette er det viktig at sikkerheten blir ivaretatt også på denne trådløse kortdistanse-teknologien. Dette innebærer såkalt "paring" som de fleste Bluetooth-brukere har gjort seg bekjentskap med. Å pare går ut på å koble sammen to enheter ved hjelp av en delt nøkkel[24].

Denne nøkkelen kan en bestemme selv. Men bare de mer avanserte enhetene. Et Bluetooth headset for eksempel, blir ikke regnet som en avansert enhet og benytter seg derfor av en egen nøkkel som ikke kan endres.

Bluetooth har flere bruksområder, blant annet:

- Overføring av filer
- Snakke gjennom trådløse headset
- Kobling av trådløse tastatur og mus med datamaskin
- Printer-tilkobling
- Etc.

For at enhetene skal kunne snakke med hverandre brukes aktuelle profiler. Dette er profiler som sier noe om hvordan de forskjellige enhetene skal kobles opp mot hverandre[25]. En mobiltelefon vil typisk støtte flere av disse profilene ettersom en mobiltelefon har et større bruksområde enn for eksempel et headset som bare skal brukes til en ting.

Det finnes flere versjoner av Bluetooth. Våre testtelefoner bruker ikke den samme versjonen.

HTC HERO bruker Bluetooth versjon 2.0.[26]
Nexus One bruker Bluetooth versjon 2.1. [27]

En stor forskjell på versjon 2.0 og versjon 2.1 er at 2.1 støtter såkallt "Secure Simple Pairing (SSP). [28]

Flere angrepsmetoder mot Bluetooth har blitt definert. Følgende er hentet fra nettsiden www.bluetomorrow.com[29].

- Bluejacking
  - Ikke direkte et angrep, men en metode for å finne ut om brukeren bak en spesiell enhet er tilstede. En sender da forespørsler som brukeren må godkjenne eller ikke.
- Bluesnarfing
  - Denne angrepsmetoden gjør det mulig for en angriper å rett og slett styre en annens mobiltelefon. Sårbarheten som derimot gjorde dette mulig skal bare foreligge på eldre mobiltelefoner.
- Bluebugging
  - Denne angrepsmetoden har i prinsippet samme resultat som Bluesnarfing. En går derimot frem på en annen måte for å få tilgang.

  Utover disse angrepsmetodene eksisterer jo også DOS (Denial of Service)-angrep. En vil ikke oppnå mye ved å utføre dette angrepet, uten at offerets mobiltelefon vil bli raskere tom for strøm og kanskje ikke vil kunne bruke sin Bluetooth-tilkobling til noe fornuftig.

5.6. Kontantkortabonnement

For noen år tilbake hadde kriminelle et bra verktøy for å kommunisere med hverandre, nemlig kontantkortabonnement. Dette var abonnement som en kunne kjøpe anonymt i butikken rundt hjørnet. Konsekvensen av dette var at kriminelle kunne bruke telefon uten å bli sporet. I 2006 ble det derimot lovfestet at slike abonnement skulle kunne spores tilbake til en entydig person. Paragraf 6.2 fra EKOMforskriften omhandler dette, se under.

§ 6-2. *Informasjon om sluttbrukere*

"Tilbyder av offentlig telefontjeneste skal føre oversikt over enhver sluttbrukers navn, adresse og nummer/adresse for tjeneste. Oversikten skal inneholde opplysninger som muliggjør entydig identifisering av de registrerte og opplysninger som muliggjør geografisk lokalisering av de registrerte i forbindelse med nødanrop, jf. § 6-3 annet ledd og ekomloven § 2-6. Informasjon om offentlig betalingstelefon skal omfatte adresse"

Dette innebærer i teorien at ingen kan eie og bruke en telefon anonymt. Om dette fungerer i praksis er derimot en helt annen sak. Vi synes dette er et relevant spørsmål ettersom dette vil påvirke om kriminelle vil bruke mobiltelefonen som et redskap til å spre ondsinnet programvare. (Det vil være enklere å gi et telefonnummer til politiet enn en IP-adresse som kan ha opprinnelse på andre siden av jordkloden). Vi har kontaktet flere teleoperatører angående hvordan de håndterer registreringen av nye kontantkortabonnement. En felles prosess blant alle spurte er følgende:

Fødselsnummer og navn på kunde må oppgis. Dette blir sjekket opp mot folkeregisteret for å se om koblingen stemmer (at navn hører til fødselsnummer). Hvis dette ikke stemmer går ikke registreringen gjennom. Stemmer informasjonen, blir registreringen godtatt og abonnementet blir aktivert. Da det i disse tider finnes personopplysninger på avveie, synes vi ikke at denne sjekken mot folkeregisteret er noe stort hinder for de som virkelig er målbevisste. En vil derimot hindre Ola Nordmann fra å oppgi feilaktige opplysninger og/eller registrere en telefon på en annen person.

Etter denne prosessen begynte det derimot å sprike. Noen sendte et bekreftelsesbrev til kundens adresse for å fange opp de som hadde blitt utsatt for identitetstyveri. Dette synes vi er bra, ettersom kunden (i teorien) blir gjort oppmerksom på at et abonnement har blitt tegnet i hans/hennes navn. Hvis det ikke er kunden som har tegnet abonnementet tar vedkommende kontakt med teleoperatøren for å ordne opp. Det er derimot to store men her;

1. Abonnementet blir aktivert umiddelbart etter registrering, en kriminell har derfor et par dager på seg til å gjøre seg ferdig med sine handlinger før offeret overhode vet at dets navn har blitt brukt i et identitetstyveri. Offeret må gjennom en tung prosedyre for å vise sin uskyld, til tross for at vedkommende ikke har gjort noe galt.
2. Kriminelle blir mer og mer utspekulerte. De kjenner prosessen, og venter ved offerets postkasse. Den kriminelle fjerner så bekreftelsesbrevet når det ankommer. Man kan eventuelt endre adressen som post fra teleoperatøren blir sendt til, men dette gjør at en blir mindre anonym.

Punkt 2 forutsetter at den kriminelle både vet hvor offeret bor, og har mulighet til å komme seg dit.

En teleoperatør sendte i tillegg til overnevnte prosedyre, også en e-post med all informasjon.

En annen teleoperatør sendte i stedet for bekreftelsesbrev til folkeregistrert adresse, en SMS. Dette anser vi som et hull i deres prosedyrer i forhold til de andre operatørene. Dette for en ikke forebygger identitetstyveri. En kriminell kan registrere et abonnement på andre siden av landet, uten at offeret finner noe ut. Dette gjelder også de teleoperatører som ikke gir ut noen form for dokumentasjon på at registrering har blitt foretatt.

Vår konklusjon er derfor at EKOMforskriften ikke hindrer kriminelle fra å "legge skylden på andre", men har lagt terskelen for å gjøre dette mye høyere.

Teleoperatørene har i alle fall en prosess som omhandler EKOMforskriften. Om prosessene er gode nok i forhold til lovverket lar vi det være opp til juristene å bestemme.

6. Testing

Dette har i stor grad vært en kartleggingsoppgave, det vil si at vi har brukt mye ressurser på research. Vi har prøvd å finne ut så mye som mulig av hva andre har kommet frem til, slik at vi ikke finner opp hjulet på nytt. To emner, Bluetooth og WIFI, er områder som vi synes det er interessant å ta en nærmere titt på, dette fordi det er standarder som det eksisterer flere angrepsmetoder på. Vi har satt opp vårt eget lille testmiljø der vi har prøvd å finne ut litt selv. Vi vil derimot presisere at vi ikke har altfor stor kompetanse på dette område og har sett det an hva vi har kunnskap om fra tidligere og hva vi har hatt tid til å sette oss inn i.

6.1. Bluetooth

Hvis en aktiverer Bluetooth på Android-telefonene, er det aktivert til en skrur det av selv. Dette er noe som er nødvendig å ha som standard ettersom en ikke vil at en tilkobling skal bli brutt før all overføringsbehov er ferdig. Som vi også skriver senere i dette emnet, er det viktig å ikke ha Bluetooth aktivert når det ikke er nødvendig. Dette gjelder både med tanke på strømforbruk og på sikkerhetsrisikoer.

De fleste Bluetooth-enheter opererer i tillegg til å ha Bluetooth aktivert eller ikke, med en funksjon som handler om synligheten til enheten. Hvis en enhet er synlig sender den ut enhetsnavnet til alle rundt slik at de enkelt kan koble seg til å pare. Den generelle "regel" er å ikke ha enheten i synlighetsmodus, nettopp på grunn av at andre kan se den og utføre angrep mot den.

En sikkerhetsmekanisme i Android er at når man setter en telefonen i synlighets-modus, vil telefonen være i denne modusen i maks to minutter, for deretter å gå tilbake i usynlighets-modus igjen. Denne usynlighets-modusen benytter flere seg av og øker sikkerheten med bluetooth betraktelig. Dette fordi andre enheter som nevnt ikke kan se Bluetooth-enheten, og derav ikke vet at den er der.

For å kunne utføre noen form for testing over Bluetooth i vårt tilfelle, må telefonen ha dette aktivert i tillegg til at enheten vi skal teste med har støtte for teknologien. I vår test vil vi se hvor mye informasjon vi kan få ut fra en Bluetooth-aktivert telefon, samt nevne mulige sikkerhetsrisikoer vi synes er viktig å merke seg.

En takk går til produsenten av følgende "how to"-film, "Underground – Bluetooth Hacking"[30]. Denne filmen viser noen av konseptene med angrep som er mulig, og hvordan en går frem. (Disse angrepene er dog stort sett foreldet med tanke på de nyere smarttelefoner).

I vår test bruker vi en laptop med BackTrack[G] - en Linux distribusjon (ofte brukt i blant annet penetrasjonstester på grunn av alle verktøyer som er lagt inn).

Vi starter med å skrive inn kommandoen *'hciconfig'* som en bruker for å konfigurere Bluetooth-enheter. (Figur 6.1.1)

```
root@bt:~# hciconfig
hci0:   Type: USB
        BD Address: 00:24:2C:FB:24:54 ACL MTU: 1017:8 SCO MTU: 64:8
        DOWN
        RX bytes:2731 acl:0 sco:0 events:88 errors:0
        TX bytes:207 acl:0 sco:0 commands:46 errors:0
```

Figur 6.1.1

Her ser vi blant annet at Bluetoothenheten går under navnet 'hci0' internt på vår laptop. MAC-adressen er 00:24:2C:FB:24:54. Denne adressen kan en derimot forandre til hva en skulle ønske. Derfor er det ikke uten videre noe stor sikkerhet hvis en skulle drive med diverse MAC-filtreringer. (Noen angrep går ut på å legitimere seg som enheten en angriper ved å bruke samme MAC-adresse).
I tillegg kan vi se at Bluetooth ikke er aktivert (DOWN). Vi må derfor først aktivere denne.

Vi skriver da inn *'hciconfig hci0 up' (Figur 6.1.2)*

```
root@bt:~# hciconfig hci0 up
root@bt:~# hciconfig
hci0:   Type: USB
        BD Address: 00:24:2C:FB:24:54 ACL MTU: 1017:8 SCO MTU: 64:8
        UP RUNNING
        RX bytes:3085 acl:0 sco:0 events:100 errors:0
        TX bytes:245 acl:0 sco:0 commands:57 errors:0

root@bt:~#
```

Figur 6.1.2

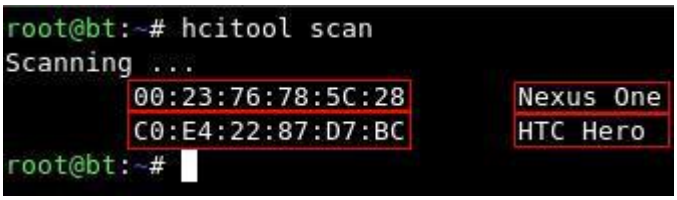
Nå er Bluetooth aktivert på vår laptop (UP RUNNING) som vist på figur 7.1.2. Nå skal vi se om vi finner noen Bluetooth-enheter i området.

For å søke etter Bluetooth-enheter med vår laptop bruker vi kommandoen *'hcitool scan'*.

Figur 6.1.3

Som en kan se dukker ikke noe opp (Figur 6.1.3). Dette regnet vi med. Begge telefonene har derimot Bluetooth aktivert, men de er satt til å være "undiscoverable" eller i usynlighets-modus som nevnt før i de foregående avsnittene. Allerede her ser en viktigheten av å ha telefonen usynlig. (En kan ikke undersøke noe en ikke vet er der.) Når telefonene er i denne modusen sender de ikke ut deres navn slik at andre enheter kan pare med de. På bakgrunn av dette i seg selv kan nok de aller fleste tenke seg til hvorfor en bør ha telefonene i denne modusen så ofte som mulig – hvis en har Bluetooth aktivert. Da kan ikke andre enkelt angripe din telefon.

Derimot finnes det anledninger der en må gjøre telefonen synlig. Dette gjelder for eksempel hvis en skal sende en fil til en kamerat/veninne. Vi gjør begge telefoner synlige før vi så bruker kommandoen *'hcitool scan'* igjen:



Figur 6.1.4

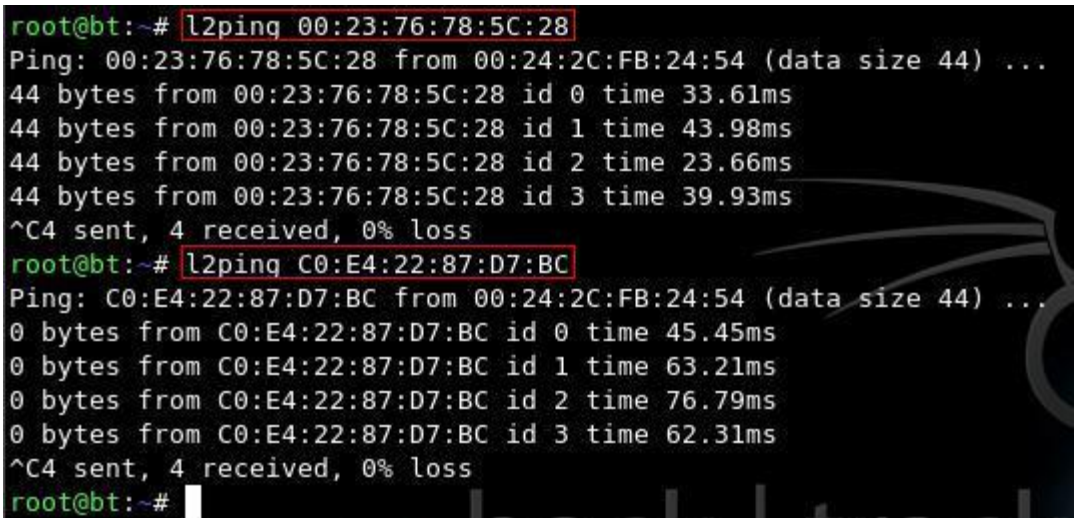
Da dukker begge telefonene opp med både egen MAC-adresse, og enhetsnavnet som blir brukt (Figur 6.1.4). Navnet endrer en selv og kan være det meste. Mer usikkert blir det derimot når noen kanskje bruker sitt fulle navn som enhetsnavn, for eksempel "Ola Nordmann". Da kan det gjøres målrettete angrep mot denne personen. Å bruke eget navn er derfor å fraråde.

Vår status nå er at vi har fått ut følgende informasjon:

Enhet 1
Navn: Nexus One
MAC: 00:23:76:78:5C:28

Enhet 2
Navn: HTC Hero
MAC: C0:E4:22:87:D7:BC

For å videre bekrefte at vi har kommunikasjon med enhetene kan vi bruke kommandoen ”l2ping”. Denne kan sammenlignes med den kjente ping-kommandoen.



Figur 6.1.5

Begge telefonene svarer (Figur 6.1.5). Det som er oppsiktsvekkende, dog logisk i forhold til hvordan Bluetooth er bygd opp, er at telefonene ikke trenger å være i synlighets-modus for at de skal svare på ping. Dette betyr at hvis en angriper får tak i MAC-adressen til en telefon, kan telefonen angripes så lenge Bluetooth er aktivert. I et slikt scenario, der angriper har tilgang til MAC-adressen, vil usynlighets-modus være ubrukelig, for angriperen vet allerede at telefonen er tilgjengelig.

For å hindre slikt fra å inntreffe anbefaler vi derfor å gjøre det vanskeligere for angriperen å få tak i MAC-adressen i første omgang. Sørg for å være alene når en setter telefonen i synlighetsmodus. Hvis en slipper å gjøre telefonen synlig i offentligheten, minimerer en sjansen for at noen får tak i din MAC-adresse.

Gjennom vår research har vi kommet over noen personer fra University College London som har forsket på muligheter for å hente ut informasjon fra en Bluetooth-enhet som ikke er synlig[31]. Noe av utfordringen på dette emnet er at Bluetooth-enheter bedriver såkalt ”frequenzy-hopping” som gjør at de hopper fra frekvens til frekvens, 1600 ganger i

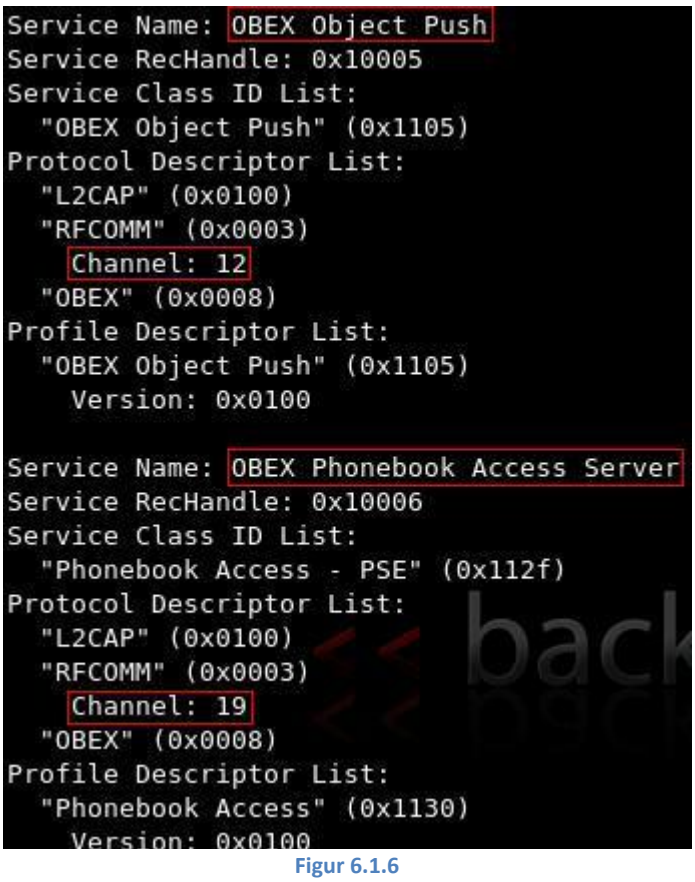
sekundet. Dette for å hindre interferens og for å gjøre det vanskeligere å bryte seg inn i en Bluetooth-tilkobling. Rekkefølgen i hoppene er ikke satt. Det vil si det er ikke en fast rekkefølge hoppene blir gjort. Hoppene blir kalkulert ut fra en algoritme som bruker informasjon fra enhetene, for eksempel MAC-adresse.

Disse personene har derfor vist at det (i teorien) er mulig å finne blant annet MAC-adressen, uten at en telefon noensinne går i synlighets-modus. Vi er her inne på et relativt kjent begrep, nemlig ”Security by obscurity”. Begrepet går ut på at en prøver å sikre noe ved å skjule data. Hvis dette er tilfellet, at en kan finne MAC-adresse uten at en må gjøre seg synlig, må en gå ut ifra dette når en utvikler Bluetooth videre. En må gå ut fra at MAC-adressen alltid er kjent og derfor ikke basere sikkerheten rundt/på dette.

En kan jo også bruke en bruteforce-tilnærming for å finne en MAC-adresse. En pinger da for eksempel alle mulige kombinasjoner av MAC-adresser for å se hvilke som svarer. Dette vil derimot være svært tidkrevende.

Vi vil videre bruke en kommando som heter sdptool. Denne brukes for å gjøre queries, eller spørringer mot Bluetooth-enheter.

Ved å skrive *’sdptool browse* ”MAC-adresse”’ får vi opp alle profiler som aktuell Bluetooth-enhet støtter. Vi viser ikke alle profilene som kom frem her, ettersom det ville blitt mye bilder. Vi har trukket frem to profiler fra Nexus One slik at en kan se hvilken informasjon som foreligger (Figur 6.1.6). Dette er OBject EXchange -profilene.(OBEX Object Push og OBEX Phonebook Access Server). Ettersom de blir brukt for å overføre filer, er ikke disse å finne på HTC HERO, som ved standard oppsett, ikke har mulighet for fil-overføring.



Figur 6.1.6

Som en kan se, får vi ut bra med informasjon til tross for at vår laptop og våre testtelefoner ikke er godkjente for hverandre. Det vil si, deres forhold er helt likt som i et typisk angriper-/offer-scenario.

En angriper vil kunne få ut samme informasjon som oss. Ved å vite hvilke profiler en enhet støtter, vet man også hvilke angrips-vektorer en kan benytte seg av. Hvis man vet at det foreligger en svakhet i OBEX-profilen som kan utnyttes, gis det her informasjon som gjør at en kan koble seg direkte til denne. (Hver profil

kjører på en angitt kanal, som også står ved kjøring av 'sdptool browse').

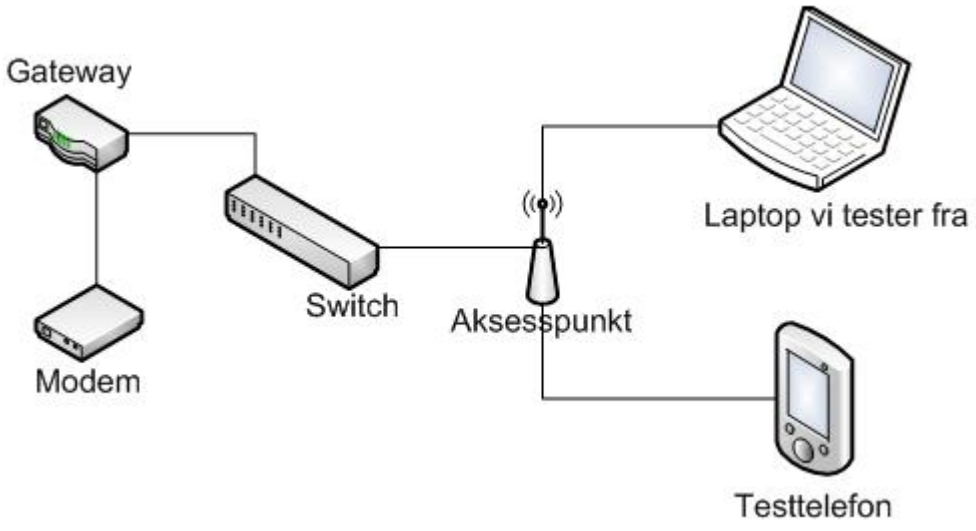
Vi har gjennom dette fått et innblikk i hvilken informasjon en kan få ut gjennom Bluetooth, ved å bruke enkle metoder.

6.2. Sårbarhetstesting- WIFI

Når det skal kjøre en sårbarhetstest på en mobiltelefon, brukes i korte trekk samme metode som på datamaskin. Mobiltelefoner er mer laget for å bare være klienter. De kobler seg oftest til server-tjenester og har få eller ingen egne server-tjenester som lytter på bestemte porter. På grunn av dette ser vi det som liten sjanse for at vi får noe stort ut av en sårbarhetstest. Vi synes allikevel at det er på sin plass å gjennomføre dette for å få et oversiktlig bilde. Om det ikke eksisterer svakheter som muliggjør kontroll over telefonen, kan det allikevel være mulig at den har svakheter som åpner opp for andre typer angrep, som for eksempel denial of service (tjenestenekt).

Vi vil bruke NMAP (ZENMAP) 5.21 for å gjøre selve portscanningen, dette er et program som blir flittig brukt blant flere som foretar sårbarhetstest.

Vårt testmiljø er hjemmenettverket til en av gruppens medlemmer. Mobiltelefonen blir koblet trådløst til et aksesspunkt (WIFI) som så er koblet i en switch. Denne switchen er koblet til en gateway, som så er koblet til modem og internett. (Se figur 6.2.1)



Figur 6.2.1

Vi vil foreta denne sårbarhetstesten med mobiltelefonen koblet til internett fordi vi anser dette som det mest brukte scenarioet (vi tror fåtallet kobler seg til et trådløst nettverk med

mindre en har som formål å komme seg på internett). Mobiltelefonen er tilbakestilt til fabrikkinnstillinger for å få et bilde over slik den er når den kommer rett fra leverandør.

Ved forespørsel om GPS-lokasjon gjennom WIFI sa vi ja til dette ved konfigurering i oppstarten.

Vi velger en av NMAP sine forhåndskonfigurerte scan-innstillinger: Intense scan, all TCP port.

Resultatet av portscan mot HTC HERO var som følger: (Figur 6.2.2)

ports

The 65519 ports scanned but not shown below are in state: **closed**

Port		State	Service	Reason	Product	Version	Extra info
3540	tcp	filtered		no-response			
16650	tcp	open	tcpwrapped	syn-ack			
27268	tcp	filtered		no-response			
31504	tcp	filtered		no-response			
32388	tcp	filtered		no-response			
33735	tcp	filtered		no-response			
34288	tcp	filtered		no-response			
35571	tcp	filtered		no-response			
36923	tcp	filtered		no-response			
49292	tcp	filtered		no-response			
49405	tcp	filtered		no-response			
50322	tcp	filtered		no-response			
52731	tcp	filtered		no-response			
57579	tcp	filtered		no-response			
59438	tcp	filtered		no-response			
63529	tcp	filtered		no-response			

remote operating system guess

used port 16650/tcp (open)
used port 1/tcp (closed)
used port 32334/udp (closed)
os match: **Android 1.5 (Linux 2.6.27)**
accuracy: 100%
reference fingerprint line number: 25515

Figur 6.2.2

Vi synes dette resultatet var overraskende, ettersom vi hadde en hypotese om at telefonen ikke ville ha noen porter åpne. Når en port blir satt som ”filtered” i en TCP-scan er dette fordi en ikke mottar noe svar fra enheten en portscanner. Det er ofte fordi en har implementert brannmur-løsninger som dropper datapakker den finner som ulegitime. For eksempel kan en regel være å forkaste alle nye inngående tilkoblinger. Da vil brannmuren forkaste datapakker som blir forbundet med en ny tilkobling og som bare har SYN-bitet satt. Da oppnår en et miljø der en enhet bare kan fungere som en klient, men ikke som en server. Vi fant det som lite trolig at vår testtelefon HTC HERO hadde en slik tjeneste implementert. Dette på grunn av at vår research ikke har sagt noe om dette, og fordi resultatet da burde ha hatt langt flere porter satt som filtered. Vi reagerer også på at portnumrene er tilsynelatende tilfeldige og det faktum at det sjelden kjøres server-tjenester i lytte-modus på

disse høgnummererte portene (servertjenester blir som oftest kjørt i den privilegerte port-rangen. (1-1024). Vi vurderte forskjellige grunner for hvorfor vi fikk dette resultatet og kom frem til at det trolig var fordi enheten eller testmiljøet ikke klarte å prosessere alle de forespørslene som kom inn, og derfor ble hengende etter.

Vi kjørte portscannen en gang til, bare med en mindre aggressiv timing (lengre tid mellom hver pakke sendt) for å finne ut om det var noe i det vi tenkte. Dette er resultatet vi fikk: (Figur 6.2.3)

ports

The 65534 ports scanned but not shown below are in state: **closed**

Port		State	Service	Reason	Product	Version	Extra info
16650	tcp	open	tcpwrapped	syn-ack			

remote operating system guess

used port 16650/tcp (open)
used port 1/tcp (closed)
used port 30030/udp (closed)
os match: **Android 1.5 (Linux 2.6.27)**
accuracy: 100%

Figur 6.2.3

Dette viser at det var pakketap som var grunnen til alle de filtrerte portene. På grunn av dette sto vi igjen med et inntrykk av at telefonen var svak mot DOS-angrep, ettersom den ikke håndterer datatrafikk som en enkel portscan produserte. Vi opplevde en hendelse som bygger opp bak dette inntrykket. Etter en portscan skulle vi sjekke noen WIFI-innstillinger på telefonen, da restartet telefonen seg. Vi valgte å tro at dette hadde noe med vår scan å gjøre men dette var bare en hypotese fra vår side. Vi vurderte å utføre mer spesifikke DOS-angrep mot telefonene men kom frem til at vi ikke ville bruke tid på dette.

Vi står igjen med port 16650 (TCP) som åpen.

I tillegg klarer NMAP å gjette Operativsystemet på mobiltelefonen, samt versjonen på gjeldende kjerne:

- Android 1.5 (Linux 2.6.27)

Dette stemmer overens med informasjon som står på telefonens ”About Phone”:

- Firmware Version 1.5
- Kernel Version 2.6.277-8dd6deee

Dette er altså informasjon som en kan få ut av mobiltelefonen HTC HERO uten å ha direkte tilgang, eller vite noe angående telefonen. Dette åpner opp for at en kan utføre skreddersydde angrep rettet mot 1.5 versjonen av Android, som kanskje ikke er mulige i tidligere/senere versjoner, som for eksempel 2.1. I tillegg kan svakheter være kjent med akkurat denne kjerne-versjonen. Nå må det også sies at denne type portscanning på ingen måte er anonym eller forsiktig. Mobiltelefonen blir bombardert med datapakker som lett

kunne ha bli sett hvis en hadde brukt en pakke-sniffer på telefonen. Telefonen som standard viste derimot ikke på noen måte at den var under portscanning, følgelig vil et offer ikke vite om han blir scannet eller ikke. Vi kan legge til at ved kjøring av samme portscan mot Nexus One fikk vi ikke noen resultater. Vårt fokus ble derfor HTC HERO med tanke på denne delen av rapporten (WIFI).

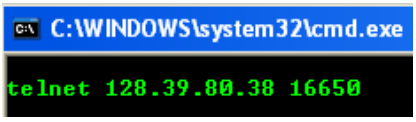
Ettersom port 16650 var åpen fikk denne videre fokus. Vi ville veldig gjerne finne ut hvilken tjeneste (hvis noen) som kjørte på denne porten.

NMAP sier at tjenesten *TCPWRAPPER* kjører på denne porten.

TCPWRAPPER

TCPWRAPPER er ment å være en ekstra sikkerhet for server-applikasjoner som kjører på en maskin (LINUX). Det som skjer er at en daemon er i listening modus på en bestemt port. Når tilkoblinger blir foretatt, blir IP til senderen sjekket opp mot en whitelist, altså IP-adresser som er godkjent for tilkobling til denne porten. Hvis IP-adressen blir godkjent, blir en server-tjeneste startet, og tilkobling til denne tjenesten blir etablert. For å være sikker på at det er TCPwrapper som kjører på denne porten må vi foreta flere tester. (Vi vet derimot ingenting om hvilken tjeneste TCPwrapper vil starte hvis IP blir godkjent)

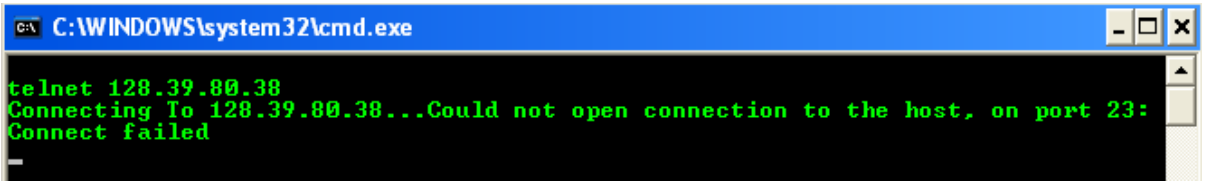
For å finne ut om det faktisk var en tjeneste som kjørte her, og hvilken, koblet vi oss til telefonen på IP-adresse og port 16650 gjennom telnet. (Denne delen ble gjennomført på Høgskolen, derav den offentlige IP-adressen. Vi anser ikke dette som noe som kan gi uriktige resultater.) Se Figur 6.2.4.



Figur 6.2.4

Ved kjøring av denne kommandoen går telnet over i ”dialog-vinduet”. Her kan en sende og motta data fra enheten en kommuniserer med. Det er derimot ikke mulig å sende noe data til enheten. Det som skjer da er at telnet lukker dette dialog-vinduet.

Vi kjørte etterpå samme kommando, bare med port 23 (en port som ikke ble flagget som åpen, dette er standardporten i telnet, og vises derfor ikke på Figur 6.2.5 under)



Figur 6.2.5

Telnet oppfører seg altså annerledes ved tilkobling til port 23 (som vi antar er stengt) og port 16650(som vi vet er åpen). Vi konkluderer derfor med at det er en tjeneste som kjører på port 16650.

Vi har brukt en del ressurser på å finne ut mer informasjon om hva denne porten 16650 (TCP) egentlig blir brukt til, uten hell. Vi har blant annet sjekket opp i flere databaser som omhandler portnumre og hva de blir brukt til. Porten fremgår som ukjent med tanke på at den er åpen på dette systemet. Vi har prøvd å etablere kontakt med ressurs-personer på noen kjente Android chatterom. Dessverre uten hell her også. Gjennom Google kunne vi heller ikke finne stort. Det er faktisk tilsynelatende få personer som har stilt dette spørsmålet, noe som vi anser som rart. Det blir ofte foretatt omfattende tester rundt dette med sikkerhet på slike morgendagens duppedingser. Det har det også blitt gjort, bare ikke på dette emnet.

Hvis en laster ned og bruker en terminal-emulator på telefonen, er det mulig å kjøre en kommando, 'netstat', som viser aktuelle tilkoblinger. Denne viser at det er noe i lytte-modus på "0.0.0.0 : 16650". Denne ligger der til tross for at en skrur av alt av trådløst kommunikasjon på telefonen.

Vi står derfor igjen med at vi har en port som er åpen på port 16650. Vi har ikke funnet ut hva som kjøres på denne porten, men tviler på at den utgjør noen stor sikkerhetsrisiko, ettersom den er så enkel å oppdage. Av prinsipielle årsaker vil en derimot når en arbeider med sikkerhet gjerne vite 100% hva en tjeneste innebærer. Utover dette opplevde vi at telefonen var potensiell svak mot stor nettverkstrafikk.

Gjennom vår portscan opplevde vi også til tider at telefonen sluttet å svare, som om WIFI-delen ble skrudd av. Etter nærmere research på dette fikk vi det til å stemme med følgende:

- WIFI blir skrudd av når skjermen går i strømsparingsmodus (skjerm blir skrudd av)
- WIFI blir skrudd av når telefon er inaktiv i lengre tid.

Vi tenker ikke på disse to hendelsene som direkte sikkerhetsmekanismer, men heller som strømsparingsmekanismer. Derimot opplever en sikkerhets-gevinster som bonus. Dette gjør i teorien at en telefon som ikke blir brukt aktivt, til tross for at den er mottakelig for SMS og samtaler, ikke kan angripes 24 timer i døgnet.

6.3. SMS Java-applikasjon

Vi bestemte oss for å snekre sammen et lite program som vi kunne bruke i presentasjonen av prosjektet for å vise hva som kunne være mulig å gjøre på telefonen hvis man fikk tilgang til den. Vi ville da lage et malware program ettersom vi ikke har funnet noen spesielle sårbarheter vi kunne utnytte. Vi så for oss at et program som kan gjøre noe med SMS ville

ha størst potensial, siden man da vil ha mulighet til å sende informasjon vekk fra telefonen uten at den er koblet til internett.

Vi valgte å lage et program som ligger og kjører i bakgrunnen uten at brukeren av telefonen vil merke det og som videresender alle innkommende meldinger til et valgt telefonnummer. Altså en liten funksjon i et såkalt spionprogram som man kan få til i de fleste operativsystem. Med tanke på at det ikke er noen form for godkjenning av applikasjoner som legges ut på Android Market[17], så kan i teorien alle programmer ha en skjult funksjon. Den eneste måten malware blir fjernet fra Android Market er ved at en bruker oppdager at programmet er malware og rapporterer dette. I dette tilfellet vil det nok ikke være så vanskelig å oppdage, med tanke på at programmet videresender alle innkommende meldinger. Brukerens telefonregning vil da nesten fordobles når det gjelder SMS, og hvis han/hun har spesifisert telefonregning vil brukeren kunne se hvilket telefonnummer som mottar alle disse meldingene. Programmet har også et eget ikon som ligger blant alle de andre installerte applikasjonene slik at det ikke vil være mulig å unngå å oppdage at det ligger installert, hvis noen skulle kunne få tak i applikasjonen vår. Vi har ikke lagt opp programmet på Android Market, og vil heller ikke gjøre dette i fremtiden.

Siden alle Android applikasjoner er laget i Java, måtte vi sette oss inn i dette språket da vi ikke kunne dette fra før av. Vi har god kjennskap til C/C++, og Java er ikke så veldig annerledes syntaks-messig, men dette tok allikevel opp litt tid. Vi måtte også sette oss inn i Android SDK[H]. Vi fulgte ikke noen spesiell metodikk når vi skulle lage dette programmet siden vi verken kunne det vi skulle gjøre eller helt visste hva vi ville ha muligheten til å lage. Vi startet hele prosessen med en liten ide om å lage et program som kunne ligge i bakgrunnen og manipulere SMS. Vi kan vel si at applikasjonen ble til etter hvert som vi lærte mer og fikk mer kunnskap om både Android's rammeverk og Java. Vi har gått igjennom mange guider og forumer for å finne ut hvordan vi skulle kunne gjøre noe i programmet. Hele programmet ble utviklet under "Code and fix" hvor vi legger inn litt kode, tester det, og enten går frem eller tilbake derfra. Vi opplevde enkelte stopp underveis hvor vi ikke fant ut hvordan vi skulle gjøre ting, og måtte da få hjelp av noen Java-eksperter og Android Community for å løse opp i problemer som egentlig var små bagateller med syntaks siden vi ikke var vant med å kode i Java. Dette tok litt tid, men vi fikk det til tilslutt.

Den endelige applikasjonen har muligheten til å snappe opp innkomne SMS-meldinger og sende de videre som SMS-melding til et forhåndsinnlagt telefonnummer med innholdet av den innkomne meldingen samt hvilket telefonnummer den kom fra. Dette kan sees på bilder videre i rapporten.

Vi vurderte å legge inn funksjonalitet for å sende videre de utgående meldingene også, samt å heller sende de over GPRS/3G for å lettere unngå å bli oppdaget når man ser på telefonregningen. Dette ville innebære at vi måtte kunne skru på og av mobilt nettverk, og

[H] Android Software Developement Kit

ville kreve mye mer tid til å kode. Dette var ikke relevant siden vi bare skulle lage et program som et konseptbevis på hvor enkelt det er å lage en simpel malware. Hvis programmet installeres på en telefon, vil det fortsatt ligge et ikon i programmenyen. Dette fordi vi enkelt skal kunne starte programmet på nytt og fordi vi ikke vil skjule det helt. Hvis vi skulle skjult programikonet hadde det kunnet bli gjort ved bare å fjerne en linje med kode i kildekoden.

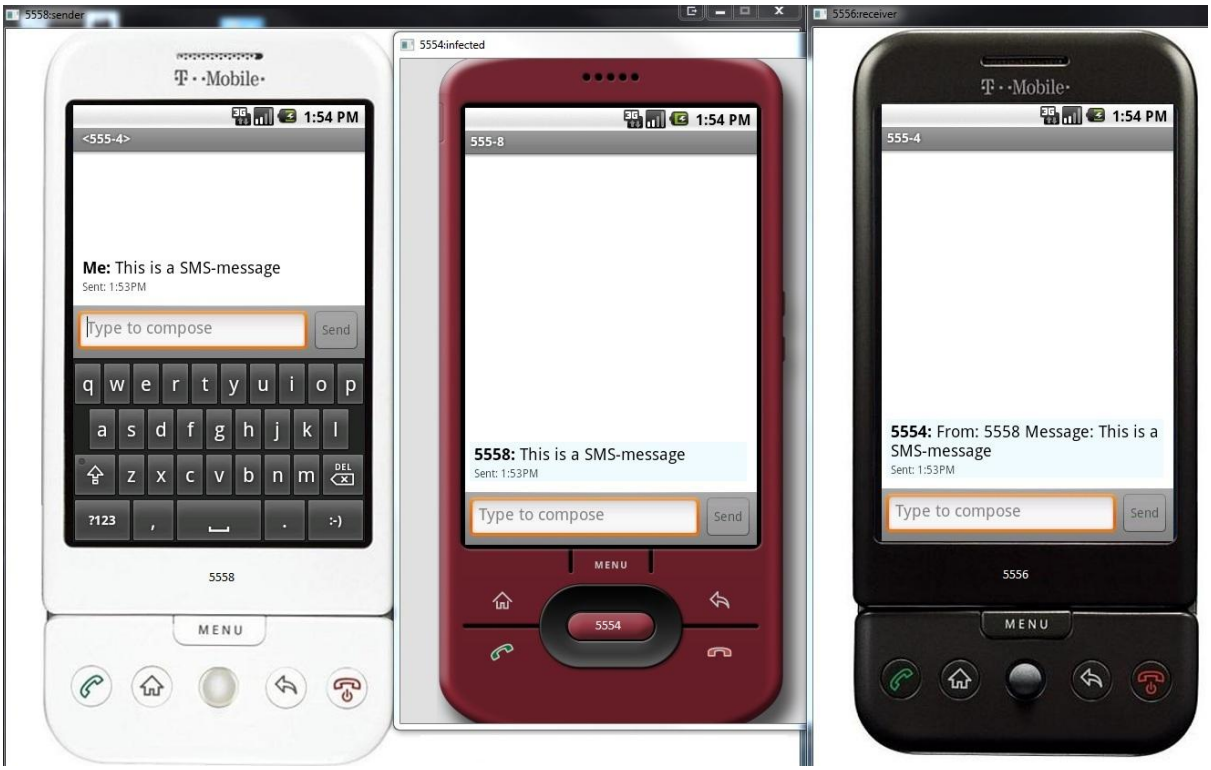
Nedenfor har vi tre bilder som illustrerer vår applikasjon. De forskjellige telefonene representerer brukere:
Hvit telefon: Hvilken som helst bruker som sender melding til rød telefon.
Rød telefon: Den infiserte telefonen.
Svart telefon: Kriminell sin telefon, mottar meldingene sendt fra den røde telefonen.

Figur 6.3.1

Programmet er installert og startet på den ”infiserte” telefonen. (Figur 6.3.1)

Figur 6.3.2

Sender en SMS til den infiserte telefonen. (Ettersom dette er emulert på en datamaskin, vil portnummer operere som telefonnummer. ”5554” er altså telefonnummeret til den røde, infiserte telefonen. (Figur 6.3.2)



Figur 6.3.3

SMS'en blir videresendt til telefonnummeret som ble valgt ved første oppstart av programmet. Den sorte telefonen mottar meldingen og får da opp hvilket nummer meldingen ble sendt fra i første omgang og hva meldingen inneholder. (Figur 6.3.3)

Vi testet først i emulatoren (som på bildene) at dette skulle fungere prikkfritt, og gikk videre for å installere det manuelt på vår testtelefon HTC Hero. For å få installert programvare som ikke blir lastet inn fra Android Market måtte vi først aktivere for installasjon av slike applikasjoner på telefonen: "Innstillinger → Applikasjoner" og huke av for "Ukjente kilder". Etter vi hadde gjort dette sendte vi over installasjonsfilen til telefonens minnekort over USB, installerte det og skrev inn telefonnummeret vi ville at meldingene skulle videresendes til. Vi testet så dette live, og alt fungerte som det skulle. Vi har kalt programmet "SMS Spy". Kildekoden til dette programmet finnes som vedlegg B.1-6.

7. Mobilbank

Etter å ha sjekket opp flere banker på nettet, og mulighetene de har for mobilbank virker det som at det ikke finnes noen form for standard på hvordan dette skal fungere. Bankene praktiserer vidt forskjellige prosesser.

Med tanke på autentisering benytter flere seg av fødselsnummer, engangskode fra kodebrikken for BankID og kontofonkode.

Utover dette benytter bankene seg av et stort repertoar for å gjøre tjenesten "sikrere", dette kan innebære for eksempel:

- Benytte engangskode fra BankID-kodebrikke før betalinger og overføringer utføres
- Maksbeløp for overførsel
- Forhåndsetting av hvilke kontonummer en kan overføre til
- Bare mulighet for å overføre til kontoer i samme bank
- Etc.

Som ved pålogging på nettbank, vil en kunne logge seg på mobilbank ved å bruke noe en har (engangskoden) og noe en vet (kontofonkode). Dette kan i flere tilfeller være en sikkerhetsrisiko. Vi tenker oss at det er på tide å innføre også på dette området en autentisering i form av noe en er. Dette kunne for eksempel vært fingeravtrykk-scanner. Dette vil øke sikkerheten betraktelig, men også da gi et krav om hardware som ikke er vanlig å ha på mobiltelefoner.

Utover dette ser vi en trend blant bankene som vi håper er godt nok gjennomtenkt. Dette er at flere banken tilbyr å vise saldo på kontoer bare ved å oppgi kontonummer og en kode, eller fødselsnummer og en kode. En trenger derimot å autentisere seg videre hvis en for eksempel skal overføre penger eller lignende, men vår bekymring angående dette foreligger fremdeles. Vi har verken fått avkreftet eller bekreftet følgende teori:

Bruteforcing av koder har lenge vært kjent innen sikkerhetsbransjen. Vi tviler derimot på at det er mulig å ta for seg et gitt kontonummer og gjøre en bruteforce på alle mulige koder som kan være forbundet med dette kontonummeret. Dette fordi det stort sett er satt restriksjoner på slik "oppførsel" ved pålogging gjennom autentiserings-tjenester. Det vår teori derimot setter spørsmål ved, er om det er satt restriksjoner på å bruke samme kode, men bruteforce på kontonummer. Det vil si at en tar for seg en kode, og prøver alle mulige kombinasjoner av kontonummer. Hvis dette er tifelle åpner det i teorien opp for at en kan finne koden til alle kontonumre, samt tilhørende saldo, i banken som tilbyr denne tjenesten. (Kontonumrene innenfor en gitt bank er satt ved de fire første sifrene i et kontonummer, disse vil derfor være statiske gjennom selve bruteforcingen). Ved en slik kartlegging kan en etterpå rette seg mot de kontonumre som har høyere saldo enn et visst beløp, og starte videre målrettete angrep gjennom for eksempel sosial manipulering.

Vi håper at overnevnte teori har blitt vurdert og at det er satt inn mekanismer for å hindre dette. Derimot ser vi også utfordringen i å hindre en slik metode, spesielt hvis det også på mobiltelefoner blir populært med såkalte "botnet". En mekanisme som kunne gjort det vanskeligere å gjennomføre et slikt angrep, er å bare la et visst forhåndsregistrert telefonnummer aksessere et kontonummer.

8. Spionprogrammer og overvåking av mobiltelefon

Det finnes utallige spionprogrammer som lar andre overvåke mobilen din. Du kan bli overvåket gjennom at man kan lese sms-meldinger (også de som er slettet), se gps-posisjon, ta bilder med kamera, sende meldinger eller ringe dyre numre, avlytte samtaler og også aktivere mikrofonen og avlytte alt som skjer i nærheten av mobiltelefonen[32].
For at du skal kunne bli overvåket må personen som skal gjøre dette ha fysisk tilgang til telefonen. Det er som oftest de nærmest deg som ville hatt muligheten til å installere et slikt spionprogram. Programmene tar bare et minutt å installere, men når det er installert er det skjult for brukeren så lenge man ikke ser igjennom listen av kjørende prosesser med et tredjepartsprogram og får identifisert dette. Når spionprogrammet er installert kan man overvåke alt telefonen blir brukt til[33].
For å få fjernet et slikt program må man som oftest enten benytte seg av et antivirusprogram, eller gjenopprette telefonen tilbake til fabrikkinnstillinger.

Det beste er å forhindre at overvåking av din mobil kan skje, ved å tenke sikkerhet fra begynnelsen av. Siden man er nødt til å installere disse programmene rett på telefonen, ikke via sms eller lignende, så vil første skritt være å aktivere en passordbeskyttelse på telefonen i tillegg til pin-kode for simkortet. Dette er altså en kode man må taste inn hver gang man vil skru av telefonens tastelås. Det kan også være greit å aktivere tidsinnstilt tastelås, slik at denne vil bli aktivert etter en viss tid med inaktivitet, hvis du har glemt å aktivere den manuelt. Neste skritt vil være å ikke legge telefonen fra seg steder hvor andre vil kunne få tak i den. En god regel vil kunne være og alltid ha telefonen på kroppen eller under oppsyn.

9. Flere funksjoner - flere sikkerhetsrisikoer

De fleste brukere i dagens samfunn har kjøpt seg verktøy som de ikke utnytter til dets fulle potensial. Dette kan for eksempel være en mikrobølgeovn som de aldri bruker tinefunksjonen på eller en forsterker de aldri bruker hdmi-inngangene på. Dette gjelder i stor grad alle varer som tilbyr flere funksjonaliteter. Det kan tenkes at dette er fordi leverandører av disse produktene vil treffe en bredere målgruppe, i tillegg til at den ekstra funksjonaliteten koster lite å legge inn sammen med hoved-funksjonaliteten. Hvis mikrobølgeovnen hadde blitt solgt med selve varmefunksjonen for seg og tinefunksjonen for seg, hadde antakeligvis markedet skreket om at de ville hatt disse funksjonalitetene i en boks, ikke to. Dette kan også relateres over til smarttelefoner. Som nevnt før kan den sammenlignes med en liten datamaskin, og tilbyr derfor de fleste tjenester/funksjoner som en datamaskin gjør.

Ettersom en datamaskin tilbyr nærmest uendelig med funksjoner og kan brukes til det meste, kan også som nevnt en smarttelefon dette. Det betyr at når man kjøper en slik telefon, er sjansen stor for at en ”får noe med på kjøpet” som en ikke trenger. Dette kan utgjøre en sikkerhetsrisiko. For eksempel kan det i noen bedrifter være svært konfidensiell informasjon som ikke kan lekke ut til omverdenen. Denne bedriften vil antakeligvis ha en policy på hvilke telefoner som skal brukes jobb sammenheng. Disse telefonene vil nok også i tillegg være forbudt å bruke privat for å minske risikoen for at noe ugunstig skjer med de. En innfallsvinkel kan være at man kjøper en smarttelefon med gode ressurser, men bare bruker telefonen for å ringe med, sende meldinger og synkronisere kalender gjennom bluetooth. Telefonen vil ha uttallig flere funksjonaliteter som for eksempel kamera og trådløst nett.

Kriminelle som vil angripe en bedrift kan få installert programvare på bedriftstelefonen som for eksempel tar bilde hvert sekund, eller minutt, for å prøve å skaffe seg tilgang til konfidensielt materiale. Alternativt kan de ta opp en pågående konfidensiell samtale for å hente ut kritisk informasjon. Tiltak for å hindre slike hendelser kan være å deaktivere selve

funksjonene slik at det verken er mulig å ta bilder eller ringe noen. Kanskje det er nok for brukeren av telefonen å bare ha mulighet for å sende meldinger?

Per dags dato vet vi ikke om det er mulig å skreddersy en telefon på en slik måte. Det vil være ugunstig for en leverandør å måtte konfigurere hver eneste enhet i stil til det en kjøper trenger. Derimot er dette fullt mulig med dagens teknologi. Flere leverandører har også tatt i bruk dette, dog på andre produkter, som biler for eksempel.

Et motargument til en slik tilnærming derimot, er at kunden sjelden har full oversikt over hva han/hun trenger. Da er det enklere å gå for standardversjonen. Hvis en kunde ville benyttet seg av dette tilbudet, hvilke muligheter ville kunden hatt i ettertid for å endre på valget?

En løsning kunne vært å opprettholdt et serviceverksted der en kunne levert inn telefonen for slik skreddersying.

En annen løsning, for bedrifter som har behov for slik skreddersying, er at bedriftene får frie tøyler til å gjøre dette selv, uten at dette går ut over garanti eller lignende. Dette ville gitt fleksibilitet ettersom bedriften har førstehåndskjennskap til hva de trenger. IT-avdelingen kunne hatt ansvar for at telefonen blir konfigurert opp som tenkt, med eventuelle risikoer grunnet uvitenhet dette måtte medføre.

9.1. Funksjoner i telefonene, som innehar sikkerhetsrisiko

9.1.1. Maskinvare

Kamera
Brukes for å ta bilder/filme. Kan være en stor sikkerhetsrisiko hvis det blir brukt for å ta bilde/filme konfidensiell informasjon.

Mikrofon
Brukes for å snakke i telefonen (telefonsamtale, memo, stemmestyring). Kan være en stor sikkerhetsrisiko hvis brukt til å ta opp telefonsamtaler/møter uten godkjenning.

Bluetooth

Brukes ved bruk av headset som støtter dette, overføring av filer mellom bluetooth-enheter. En sikkerhetsrisiko siden angrep er mulig uten å måtte ha direkte tilgang på telefonen.

WIFI

Brukes for å i hovedsak ha internettilgang uten å måtte betale. Brukes på samme måte som en laptop som kobler seg på WIFI. En sikkerhetsrisiko ettersom gjeldene angrep i dataverdenen også er mulig på her. (Man-in-the-middle, sniffing av data, dos-angrep, etc.)

aGPS (Assisted global position system)
Brukes for å finne ut hvor man befinner seg i forbindelse med for eksempel et navigasjonssystem. Kan være en sikkerhetsrisiko ettersom andre kan finne ut hvor telefonen(og ofte personen telefonen tilhører) befinner seg. ”Assisted” gjør det mulig å finne lokasjon til tross for at en ikke har fungerende GPS-signal. Dette gjennom for eksempel nettverks ID’en til telenettet en befinner seg i.

9.1.2. Programvare

Nettleser
Kan sammenlignes med for eksempel Internett Explorer og Firefox. Bruker for å aksessere informasjon ”på nettet”. Hvis svakheter eksisterer i nettleseren kan disse utnyttes for å skaffe seg tilgang til telefonen. Derfor en sikkerhetsrisiko. Nettlesere må holdes oppdatert!

Android Market

Brukes for å laste ned applikasjoner til telefon. Utgjør en stor sikkerhetsrisiko ettersom en aldri vet om disse applikasjonene er legitime. En risikerer at en annen person kan overvåke alt en gjør og ta over telefonen.

10. Brosjyre om sikker bruk av mobiltelefon

Ettersom prosjektet nærmet seg slutten ble vi enig med vår oppdragsgiver om å lage en skisse på en brosjyre som skulle omhandle sikker bruk av mobiltelefon, samt hvilke mobiltelefontjenester Sparebanken Hedmark tilbyr til sine kunder. Denne brosjyren var i neste omgang planlagt at skulle bli trykket opp og gjort tilgjengelig for bankens kunder. Vi hadde en effektiv arbeidsprosess vedrørende denne brosjyren der vi i gruppa lagte et utkast som vi sendte til Sparebanken. De vurderte utkastet og ga oss tilbakemelding som vi igjen brukte som grunnlag for å lage et nytt utkast. Denne iterasjons-prosessen gjorde at vi fikk raske resultat, noe som var nødvendig med tanke på den korte tiden som var igjen. Ca to uker før innleveringsfristen 20.05.2010 viste det seg dessverre at formålet med brosjyren ikke var aktuell allikevel. Vår målgruppe med brosjyren var mobilbank-kunder, dette er kunder som i stor grad bruker internett som verktøy i forbindelse med bank-relaterte

oppgaver. En ville derfor treffe majoriteten av den aktuelle målgruppen bedre ved å ha dette interaktivt på deres hjemmeside.

Dette var noe vi hadde full forståelse for og stoppet derfor med videreutvikling av brosjyren. Vi har tatt med brosjyren som den var ved dette tidspunkt som vedlegg i rapporten for å vise i hvilken retning design-messig den var tenkt og for å vise at dette var noe som ble brukt ressurser på. Vi anser dette derimot også som en lærerik hendelse med tanke på endringshåndtering som flere prosjekter har vanskeligheter med å håndtere.

10.1. Intensjonen med brosjyren

Vår intensjon med brosjyren var å øke bevissthet om sikker bruk av mobiltelefon, og farer som kunne foreligge ved bruk.

Vi anså dette som et veldig aktuelt tema ettersom det er enkelt å bli utsatt for blant annet identitetstyveri eller bli frastrålet sensitiv informasjon fra mobiltelefonen i dag. Det er brukeren selv som har et stort ansvar her, da en uvitende bruker kan være uheldig å laste ned applikasjoner på sin telefon som gjør at den blir infisert av skadelig programvare. I et slikt tilfelle kan telefonen for eksempel bli brukt som et overvåkningsverktøy. Kamera og mikrofon kan bli aktivert slik at utenforstående kan høre og se alt som telefonen kan. Dette kan føre til svært uheldige konsekvenser enten det er i privat sammenheng eller i det offentlige. Det må derimot også nevnes at brukere som kan regnes som oppegående innen informasjonssikkerhet også kan komme til å installere slike skadelige applikasjoner på sin telefon. Vi mener derfor det er på sin plass å nå en relativ stor målgruppe med vårt budskap.

10.2. Hva med brosjyren nå?

Til tross for at brosjyren ikke blir brukt til det egentlige formål er den enda relevant for Sparebanken Hedmark. Ettersom de har planer om å legge inn mobilvett på deres hjemmeside vil de ta utgangspunkt i informasjonen i vår brosjyre når de skal skrive dette. Utkastet av brosjyren, så langt vi kom med den før det ble bestemt at den ikke skulle brukes som brosjyre, kan finnes som vedlegg A.

11. Anbefaling til banken

Ettersom vi hadde en god dialog med Sparebanken Hedmark vedrørende utviklingen av vår nåværende brosjyre, kan/bør dens innhold brukes som utgangspunkt i formidling av videre råd/tips angående mobilvett.

Vi foreslår uansett at banken arbeider videre med bevisstgjøring av risikoen ved bruk av mobiltelefoner og hva kundene skal være påpasselig med for å bruke disse enhetene sikkert, da med tanke på både bankens ansatte og kunder.

Et lignende dypdykk som vi har tatt for oss i denne rapporten kunne også vært aktuelt å gjennomføre på andre operativsystemer som for eksempel Windows Mobile, Symbian og iPhone for å få et bredere overblikk over tekniske sikkerhetsmekanismer.

Viser også til punkt nummer 13 ”Videre arbeid” hvor vi tar for oss mer generelle muligheter for arbeid videre.

12. Konklusjon

Det er flere aspekt med mobiltelefoner som kan utgjøre en sikkerhetsrisiko. De fleste har som oftest med seg sin telefon hvor enn en går. Dette kan resultere i at en blir overvåket når som helst på døgnet, enten en er på møte, i en viktig telefon eller bare ligger og sover. Brukernes sikkerhetskompetanse utgjør en stor del av forsvaret mot morgendagens angrep, og det er dette som ut ifra vår rapport er utfordringen. Informasjonssikkerhet er et omfattende område som en ikke kan forvente at den vanlige mannen i gata har satt seg inn i eller har motivasjon til å lære. Vi ser det derfor som en god løsning å praktisere bevisstgjøring der det er mulig for å få frem hvor viktig det er med sikkerhet. En kommer en god vei bare ved å tenke seg om to ganger. Utover dette vil det være særdeles smart for en bruker å anskaffe seg et anti-virus program på sin mobiltelefon. Slike programmer er til stor hjelp for å luke vekk eventuelle skadelige applikasjoner som måtte bli installert i god tro, eller ved en feiltakelse. Dette vil derimot sette en demper på prosesserings-mulighetene til enheten ettersom slike applikasjoner har en tendens til å trekke en del ressurser. Dette kan nok tenkes å være hovedgrunnen for at det ikke levereres mobiltelefoner med anti-virus som standard per i dag.

Det kan virke voldsomt å si at det er brukeren selv som må bli mer bevisst. Noe må vel leverandørene av slike enheter kunne gjøre? Sannheten, som også har vært kjent en god stund, er at sikkerhet ofte går på bekostning av brukervennlighet. For å sikre brukerne mot seg selv har for eksempel Google satt inn restriksjoner i sitt Android operativsystem som

gjør at en ikke kan operere som root/administrator når en kjøper en smarttelefon. Dette hindrer skadeomfanget hvis brukeren skulle ”utforske” litt på egen hånd. Mennesket derimot, liker frihet - og missliker begrensninger. Derfor finner personer med god kompetanse finurlige måter å unngå slike sikkerhetsmekanismer på. Resultatet blir at brukerne – som egentlig ikke er tenkt at skal ha root/administrator-rettigheter, skaffer seg dette gjennom metoder som går på tvers av den sikkerheten som Google hadde tenkt i første omgang. Vi mener det er forskjell på å utvikle et sikkert operativsystem som bygger på at brukerne ikke har disse rettighetene og å utvikle et der brukerne faktisk har de. En bruker da et operativsystem på en måte som ikke var tenkt, og dette åpner for nye sikkerhetsrisikoer.

Google Android virker altså som et veldig sikkert operativsystem, med unntak av Android Market hvor man er nødt til å være påpasselig ved å ikke installere programmer som ber om flere ressurser enn hva de faktisk trenger for å utføre de oppgavene som er beskrevet. Det er igjen her brukeren av telefonen som må tenke seg om et par ganger før installering og ikke utføre noen ”krampe-installering” hvor man bare trykker seg igjennom raskest mulig for å få programmet installert. Alt i alt så hviler de fleste sikkerhetsrisikoer på hvor opplyst brukeren av telefonen er når det gjelder sikkerhet.

13. Videre arbeid

På bakgrunn av våre funn i denne rapporten tenker vi oss flere områder det kan arbeides videre med:

- Innføring av sjekk for skadelig kode på Android Market. Trenger en applikasjon virkelig tilgang til alle de ressursene den ber om?
    - Dette vil minske antall skadelige applikasjoner nedlastet

- Utføre mer omfattende prosesser i forhold til å autentisere en person ved registrering av kontantkortabonnement (vi anser ikke lenger fødselsnummer + navn til å være nok)

- Undersøke om sikkerhetsmekanismene innenfor Bluetooth er sikre hvis MAC-adresse og enhetsnavn er kjent
    - Forskning tyder på at usynlighetsmodusen ikke vil være holdbar i alt for lang tid

- Gjennomføre en grundig sikkerhetsanalyse av Webkit (den medfølgende nettleseren i Android)

    - Dette programmet mottar sjeldent eller aldri noen oppdateringer annet enn hvis man oppdaterer hele telefonen med et nytt ROM. Som vi nevnte i begynnelsen av rapporten lanseres det veldig sjeldent et nytt rom og oppdagede sårbarheter ligger da i lang tid uten å bli fikset.

- Vurdere om en standard for å skreddersy smarttelefoner kan defineres og iverksettes. Dette kan i første omgang starte med en spørreundersøkelse på relevansen i en slik standard. Utfordringene bør kartlegges i stor grad. Konseptet kan for eksempel være at mobiltelefonen:
    - får sitt kamera/mikrofon deaktivert, men allikevel kan brukes for å ha kontakt med omverdenen, for eksempel gjennom SMS.
    - må ha aktivert tastelåskode i tillegg til kode på SIM
    - må ha innstallert et fungerende anti-virus program for å fungere

- Innføring av biometrisk autentisering på mobiltelefon
    - En videreutvikling av touchskjerm, eventuelt en egen fingeravtrykkleser på telefonen vil øke integriteten i en autentisering ettersom en i tillegg til å vise til det en vet (eksempelvis passord) og det en har (eks. selve telefonen/bankid) også kan vise til noe en ER (eks. fingeravtrykk).

14. Vedlegg

A. Utkast av Brosjyre om mobilsikkerhet
B. Kildekode for SMS Spy
   B.1. AndroidManifest.xml
   B.2. Global.java
   B.3. main.xml
   B.4. MainActivity.java
   B.5. SMSReceiver.java
   B.6. SMSSender.java
C. Evaluering av gruppens arbeid
D. Timelogg
E. Forprosjektrapport
F. Faktisk Gannt-skjema
G. Risikovurdering av prosjektet
   G.1. Per 17.02.2010
   G.2. Per 09.04.2010
H. Møteinnkallinger/referater
   H.1. Møteinnkalling 07.01.2010
   H.2. Møtereferat 07.01.2010
   H.3. Møteinnkalling 10.02.2010
   H.4. Møtereferat 10.02.2010
   H.5. Møteinnkalling 26.03.2010
   H.6. Møtereferat 26.03.2010

15. Referanser/kilder

[1] GSA, GSM/3G Stats, 2010, http://www.gsacom.com/news/statistics.php4
[2] Symbian, Symbian is Open Source, 2010, http://www.symbian.org/symbian-feature-set/symbian-is-open-source
[3] Marius Jørgenrud, Han knekker GSM-koden, 04.05.2010, http://www.digi.no/841586/han-knekker-gsm-koden
[4] Google I/O, Anatomy and Physiology of an Android, 2008, http://developer.android.com/videos/index.html#v=G-36noTCaiA
[5] David Sugar, The GNU "Lesser" General Public License gets some love, 18.07.2008, http://www.freesoftwaremagazine.com/node/1681
[6] Android Developers, Community, http://developer.android.com/community/index.html
[7] Asaf Shabtai, Yuval Fledel, Uri Kanonov, Yuval Evolici and Shlomi Dolev, Google Android: A State-of-the-Art Review of Security Mechanisms
[8] Per-Øyvind Nordberg, Kjempeoppdatering til X10, 19.04.2010, http://www.amobil.no/artikler/kjempeoppdatering_til_x10/75987
[9] Wiki artikkel, Why Root?, 2010, http://android-dls.com/wiki/index.php?title=Why_Root
[10] Jesterz, [HOWTO] If you flashed the original RUU_Hero_HTC_WWE_2.73.* and got the hboot new, 28.09.2009, http://forum.xda-developers.com/showpost.php?p=4525393&postcount=1
[11] Jackdaniels_lee, Rooting the HTC Hero/G2..(success), 16.12.2009, http://forum.xda-developers.com/showthread.php?t=543571
[12] Sayed Hashimi, Satya Komatineni & Dave MacLean, How the Dalvik Virtual Machine Works on Google Android, 08.03.2010, http://www.ctoedge.com/content/how-dalvik-virtual-machine-works-google-android
[13] Dalvik Virtual Machine, Brief overview of the Dalvik virtual machine and its insights, 2010, http://www.dalvikvm.com/
[14] First Tech, Fraud, 2010, http://www.firsttechcu.com/home/security/fraud/security_fraud.html
[15] Android Market, 2010, http://www.android.com/market/
[16] Jacob Leibenluft, Credit Card Numbers for Sale, 24.04.2008, http://www.slate.com/id/2189902/
[17] Andrew Kameka, Google bans about 1% of Android Market Apps, 25.08.2009, http://androinica.com/2009/08/25/google-bans-about-1-of-android-market-apps/
[18] Rob Jackson, Android 2.2 Allows Automatic App updates, 22.04.2010, http://phandroid.com/2010/04/22/android-2-2-allows-automatic-app-updates/
[19] Securelist, Mobile Malware Evolution: An Overview, 29.09.2009, http://www.viruslist.com/en/analysis?pubid=204792080
[20] Wiki artikkel, Malware, 2010, http://en.wikipedia.org/wiki/Malware
[21] Harald Brombach, MMS-orm truer smarttelefoner, 08.03.2005, http://www.digi.no/210307/mms-orm-truer-smarttelefoner
[22] Pål Joakim Olsen, Er det rart folk får virus?, 10.03.2010, http://www.dinside.no/837352/er-det-rart-folk-faar-virus
[23] BlueTomorrow, Bluetooth History, 2010, http://www.bluetomorrow.com/about-bluetooth-technology/history-of-bluetooth/bluetooth-history.html
[24] BlueTomorrow, Bluetooth Pairing, 2010, http://www.bluetomorrow.com/about-bluetooth-technology/general-bluetooth-information/bluetooth-pairing.html
[25] BlueTomorrow, Bluetooth Profiles, 2010, http://www.bluetomorrow.com/about-bluetooth-technology/how-bluetooth-works/bluetooth-profiles.html
[26] HTC, Hero Spesifikasjoner, 2010, http://www.htc.com/no/product/hero/specification.html
[27] Google, Nexus One Spesification, 2010, http://www.google.com/phone/static/en_US-nexusone_tech_specs.html
[28] Wiki artikkel, Bluetooth, 2010, http://en.wikipedia.org/wiki/Bluetooth#Bluetooth_v2.0_.2B_EDR
[29] BlueTomorrow, Bluetooth Security – Bluejacking, Bluesnarfing, and Bluebugging, 2010, http://www.bluetomorrow.com/about-bluetooth-technology/bluetooth-security/bluejacking-bluesnarfing-bluebugging.html
[30] Patchy, Underground – Bluetooth Hacking, 26.02.2009, http://infinityexists.com/2009/02/26/underground-bluetooth-hacking/
[31] Dominic Spill & Andrea Bittau, BlueSniff: Eve meets Alice and Bluetooth, 2010, http://www.usenix.org/event/woot07/tech/full_papers/spill/spill_html/

[32] Flexixpy, 2010, http://www.flexixspy.com
[33] Marius Blaker, Så lett overvåkes mobile din, 18.05.09, http://www.nettavisen.no/it/article2626307.ece
[34] Android Open Source Project, Dalvik, 2010, http://pdk.android.com/online-pdk/guide/dalvik.html
[35] Android-porting, Maillist, 2010, http://www.mail-archive.com/android-porting@googlegroups.com
[36] MobiStealth, 2010, http://mobistealth.com
[37] Dan Bornstein, Dalvik Virtual Machine Internals, 2008, http://sites.google.com/site/io/dalvik-vm-internals
[38] My Digital Life, How to Unlock Bootloader and Root or Jailbreak Google Nexus One, 2010, http://www.mydigitallife.info/2010/01/26/how-to-unlock-bootloader-and-root-or-jailbreak-google-nexus-one/
[39] Taimur Asad, Root Google Nexus One on Android 2.1, 31.12.2009, http://www.redmondpie.com/how-to-root-google-nexus-one-on-android-2.1-9140296/
[40] Justin Shapcott, The Dangers of rooting your Android phone, 16.09.2009, http://androidandme.com/2009/09/hacks/why-not-everyone-should-root-their-android-phones/
[41] Rob Jackson, Rooting Android: Worth it or not? 27.01.2010, http://phandroid.com/2010/01/27/rooting-android-worth-it-or-not/
[42] Shawn Brown, How to gain root access on your HTC Hero, 04.08.2009, http://phandroid.com/2009/08/04/how-to-gain-root-access-on-your-htc-hero/
[43] Uli Ries, Researchers show infecting smartphones with malware is relatively easy, 09.03.2010, http://www.h-online.com/security/news/item/Researchers-show-infecting-smartphones-with-malware-is-relatively-easy-950091.html
[44] Marius Jørgenrud, Svindel-app I Android Market, 11.01.2010, http://www.digi.no/832580/svindel-app-i-android-market
[45] SMSspoofing, Everything You Ever Wanted To Know About SMS Spoofing, http://www.smsspoofing.com/
[46] WAP and Viruses – Can your mobile phone get infected?, Mikko Hyppönen 2000
[47] Mikko Hyppönen, State of cell phone malware in 2007
[48] Job de Haas, Mobile Security: SMS and WAP, 2001 (Black Hat Amsterdam)
[49] User Oriented Machines, A Security Flaw in Google Android, 22.02.2009, http://useroriented.wordpress.com/2009/02/22/a-security-flaw-in-google-android/
[50] Security tracker, Google Android SMS and Dalvik API Bugs Let Remote Users Deny Service, 2010, http://securitytracker.com/alerts/2009/Oct/1022986.html
[51] oCert Advisories, Android denial-of-service issues, 2010, http://www.ocert.org/advisories/ocert-2009-014.html
[52] Securiteam, Android Malformed SMS and Dalvik API DoS Vulnerabilities, 2010, http://www.securiteam.com/unixfocus/6T0010UQ0K.html
[53] Google Code, Issues List, 2010, http://code.google.com/p/android/issues/list
[54] Sarah Perez, Android Vulnerability So Dangerous, Owners Warned Not to Use Phone's Web Browser, 12.02.2009, http://www.readwriteweb.com/archives/android_vulnerability_so_dangerous_shouldnt_use_web_browser.php
[55] Sean Michael Kerner, SMS, iPhone, Android Under Attack at Black Hat, 31.07.2009, http://www.internetnews.com/security/article.php/3832661
[56] SecurityFocus, Android Web Browser GIF File Heap-Based Buffer Overflow Vulnerability, 2010, http://www.securityfocus.com/bid/28005/discuss
[57] Thomas Claburn, Black Hat: Android, iPhone SMS Flaws Revealed, 29.07.2009, http://www.informationweek.com/news/security/vulnerabilities/showArticle.jhtml?articleID=218800192
[58] Plusminus/Android Development Community, Working with Files, 27.11.2007, http://www.anddev.org/working_with_files-t115.html
[59] Android Developers, How do I securely use my Android phone?, 2010, http://developer.android.com/guide/appendix/faq/security.html#use
[60] Jesse Burns, Mobile Application Security on Android, 2009 (Blackhat)
[61] Aubrey-Derrick Schmidt, Rainer Bye, Hans-Gunther Schmidt, Jan Clausen, Osman Kiraz, Kamer Ali Yuksel, Seyit Ahmet Camtepe, Sahin Albayrak, Static Analysis of Executables for Collaborative Malware Detection on Android
[62] Android Developers, Security and Permissions, 2010, http://developer.android.com/guide/topics/security/security.html

[63] Post- og Teletilsynet, Mangelfull registrering av sluttbrukere – særlig om kontantkortkunder, 2006
[64] Post- og Teletilsynet, Registrering av sluttbrukere – særlig om kontantkortkunder, 2004
[65] Arctis Developer Reference, Creating Building Blocks for Android, 2010, http://arctis.item.ntnu.no/doc/creating_building_blocks_for_android
[66] Davanum Srinivas, Android – Listen for incoming SMS messages, 15.12.2007, http://davanum.wordpress.com/2007/12/15/android-listen-for-incoming-sms-messages/
[67] Weimenglee, SMS Messaging in Android, 2009, http://mobiforge.com/developing/story/sms-messaging-android
[68] Plusminus/Android Developement Community, Recognize/React on incoming SMS, 16.12.2007, http://www.anddev.org/recognize-react_on_incoming_sms-t295.html
[69] Android Competency Center, Andrpod API – SMS Handling, 29.12.2008, http://www.androidcompetencycenter.com/2008/12/android-api-sms-handling/
[70] GSM Tech, SMS FAQ, 2010, http://www.gsm-technology.com/gsm.php/en,unlock,subpage_id,smsfaq.html
[71] Routo Messaging, Message Types, 2010, http://www.routomessaging.com/message-sms-types.pmx
[72] Collin Mulliner, Fuzzing the Phone in your Phone, Black Hat USA 2009
[73] Mikko Hyppönen, Malware goes Mobile, 2006
[74] Jan A. Audestad, Network Security, 2009

Vedlegg

Vedlegg A

Utkast av brosjyre om mobilsikkerhet

Sikker bruk av din mobiltelefon

Mobiltjenester og sikker bruk av mobiltelefon

Sparebanken 1 HEDMARK

Innledning

Før ble mobiltelefonen brukt for å ringe og sende meldinger. Nå har mobiltelefonen blitt noe helt annet. De har blitt mindre og sterkere.

Flere går nå om dagen rundt med en liten datamaskin i lomma. Dette kombinert med at mobiltelefonen ofte er skrudd på, kan i noen tilfeller være usikkert. Det er derimot i vår interesse at du som kunde kan bruke din mobiltelefon uten å oppleve ubehag. Derfor har vi utviklet denne brosjyren, slik at du som kunde ved hjelp av enkle og raske tips kan bli en sikrere bruker.

SE HVOR DU GÅR!

Det er alltid viktig å se seg for. Dette gjelder også innen telefon-sikkerhet.

Hvis du bruker din telefon uten å tenke på hvilke farer som lurer, er det stor mulighet for at du går på en smell.

Sparebanken 1 HEDMARK

Visste du at. . .

... det kan være farligere å lagre sensitiv informasjon på din mobiltelefon enn på din datamaskin?
- En har ofte mobiltelefonen med seg, og den blir sjelden skrudd av!

... hackeren som angriper din datamaskin, også kan angripe din mobiltelefon?
- En mobiltelefon er i bunn og grunn en liten datamaskin

... den generelle bruker ikke med 100% sikkerhet kan si at en applikasjon er sikker?
- Det kan i alle applikasjoner være tilleggsfunksjonaliteter som har uetiske/uærlige hensikter

... moderne funksjoner som mikrofon, kamera og GPS kan utnyttes til å overvåke deg?
- Alt som trengs er bare en skadelig applikasjon på din mobiltelefon

... i enkelte yrker har de ikke lov til å ta med mobiltelefonen inn på møter hvor det diskuteres sikkerhetsgradert informasjon
- dette på grunn av risikoen for at mobiltelefonen er kompromittert

... det er vanskelig å oppdage om DIN mobiltelefonen er kompromittert?
- slik som på datamaskinen prøver skadelig programvare å være så usynlig som mulig

... det finnes flere hundre virus som angriper mobiltelefoner?
- Har du gjort noen tiltak for å beskytte din mobiltelefon?

Sparebanken HEDMARK

Mobilvett

- Bruk tastelås!
  - Da får ikke uvedkommende enkelt tilgang til din mobiltelefon
- Skjul inntasting av din tastelåskode
  - Hvis alle vet din personlige kode hjelper det lite for hindre andre å komme inn på mobiltelefonen din
- Skru av internett og Bluetooth når dette ikke er i b
  - Dette gjør at du sparer strøm, i tillegg til at du blir mindre mottakelig for angrep
- Ikke si ja til noe du ikke har bedt om!
  - Sjansen er stor for å bli lurt
- Vær skeptisk med å klikke på lenker du mottar
  - Vær sikker på hva lenken peker til. Det kan lede til skadelig webområde
- Innstaller et sikkerhetsprogram på din mobiltelefo hvis tilgjengelig.
  - *Et antivirus program kan være effektivt til å hold mobiltelefonen din ren for virus og spionprogrammer*
- Ikke ta med mobiltelefonen inn i viktige møter
  - Er mobiltelefonen din kompromittert kan den bli b som en fjernstyrt mikrofon

Sparebanken HEDMARK

Vedlegg A

Vedlegg A

Mobiltjenester

Mobiltelefonen er et stort satsningsområde hos oss – mange av våre banktjenester finner du også på mobiltelefonen.

- SMS-bank
  - Med SMS-bank kan du sjekke saldoen og overføre penger - uansett hvor du befinner deg. SMS-bank bestiller du fra mobilen din.

- Varsling SMS
  -Kunne du tenke deg å få en sms når det er lite penger igjen på kontoen, eller når lønna kommer inn? Logg inn i nettbanken og velg Varsling SMS i menyen for å bestille tjenesten og hvilke meldinger du ønsker å få.

- Mobilbank
  - Med mobilbank kan du utføre de vanligste nettbanktjenestene fra mobilen din. Logg inn i nettbanken og velg Mobilbank i menyen for å bestille og tilpasse mobilbanken til ditt behov.

- Telefonbank
  - Kontoinformasjon og regningsbetaling - hele døgnet - uansett hvor du er.

Illustrasjonen på forsiden er hentet fra:
http://www.dancewithshadows.com/tech/mobile-phone-security-software-top-5-anti-virus-apps/

Sparebanken 1 HEDMARK

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.abberg.sms"
    android:versionCode="1"
    android:versionName="1.0.0">

        <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-
permission>
  <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>

  <application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".MainActivity"
            android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
    <service android:name=".SMSSender">
        <intent-filter>
                <action android:name="android.intent.action.SEND" />
                <action android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
    </service>
        <receiver android:name=".SMSReceiver" android:enabled="true">
          <intent-filter>
           <action android:name="android.provider.Telephony.SMS_RECEIVED" />
          </intent-filter>
        </receiver>
   </application>


</manifest>
```

Global.java

```
package com.abberg.sms;

public class Global {                                   // Global variables
        public static String message;
        public static String phonenumber;
        public static String fromnumber;
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        android:id="@+id/mainLayout"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical"
        xmlns:android="http://schemas.android.com/apk/res/android">
<TextView
        android:id="@+id/startupText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="The application will start when device will receive an SMS.
        Please enter a phonenumber for having the messages sendt to:">
</TextView>
<EditText
        android:id="@+id/txtPhoneNo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="top"
        android:numeric="integer"
        android:focusable="true"
/>
<Button
        android:id="@+id/btnOK"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="OK"
/>
<TextView
        android:id="@+id/okView"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="You can now exit the application"
        android:visibility="invisible"
/>
</LinearLayout>
```

MainActivity.java

```
package com.abberg.sms;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
        Button btnOK;
   EditText txtPhoneNo;
   TextView okView;

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);                              // draw the layout

      btnOK = (Button) findViewById(R.id.btnOK);
      txtPhoneNo = (EditText) findViewById(R.id.txtPhoneNo);
      okView = (TextView) findViewById(R.id.okView);


      btnOK.setOnClickListener(new View.OnClickListener()
      {
         public void onClick(View v) // when the OK-button is pressed
         {
          String phoneNo = txtPhoneNo.getText().toString();
            if (phoneNo.length()>0) {
                  Global.phonenumber = phoneNo;
                  okView.setVisibility(1);   // display message that its ok to close the app
            }
            else
                  Toast.makeText(getBaseContext(),
                "Please enter phone number.",
                Toast.LENGTH_SHORT).show();  // display "popup" message
         }
      });
   }
}
```

SMSReceiver.java

```
package com.abberg.sms;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.gsm.SmsMessage;

public class SMSReceiver extends BroadcastReceiver {

   @Override
   public void onReceive(Context context, Intent intent) {  // starts when a intent is recieved
(incoming sms)

        Bundle bundle = intent.getExtras();     // get information from intent
      Object messages[] = (Object[]) bundle.get("pdus");
      SmsMessage smsMessage[] = new SmsMessage[messages.length];

      for (int n = 0; n < messages.length; n++) {   // make the sms readable
         smsMessage[n] = SmsMessage.createFromPdu((byte[]) messages[n]);
      }
     Global.fromnumber = smsMessage[0].getDisplayOriginatingAddress(); // update globals
     Global.message = smsMessage[0].getMessageBody();

      Intent smssender = new Intent(context, SMSSender.class);
      context.startService(smssender); // start SMSSender
      context.stopService(smssender);  // stop SMSSender

   }
}
```

SMSSender.java

```
package com.abberg.sms;

import android.app.PendingIntent;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.telephony.gsm.SmsManager;
import com.abberg.sms.Global;

public class SMSSender extends Service {
	public IBinder onBind(Intent intent) {
		return null;
	}

	@Override
	public void onCreate(){

		String message = "From: ";
			message+= Global.fromnumber;		// create the message
string
			message+= " Message: ";
		    message+= Global.message;
		sendSMS(Global.phonenumber, message);  // sendt the message
	}

	public void sendSMS(String phoneNumber, String message) {
		PendingIntent pi = PendingIntent.getActivity(this, 0, new Intent(this,
SMSSender.class), 0);
		SmsManager sms = SmsManager.getDefault();
		sms.sendTextMessage(phoneNumber, null, message, pi, null);  // send
message


	}
}
```

Vedlegg C

Evaluering av gruppas arbeid

Organisering

I starten av prosjektperioden fikk vi tildelt et grupperom vi kunne arbeide på. Dette var noe vi anså som en viktig forutsetning for oss ettersom vi visste at vi ville ha saker av verdi som vi skulle bruke. Refererer da spesielt til våre utleverte testtelefoner.

I starten møtte vi opp rundt klokken 8-9 hver dag for å jobbe med oppgaven og fordele arbeidsoppgaver. Vi gikk flere ganger hver til vårt for å jobbe videre. Vår oppgave dreide seg i stor grad om forskning. Vi brukte derfor utallige timer på å "søke i blinde".

Vi hadde på forhånd lite kunnskap om hva som faktisk fantes av sikkerhetsrisikoer i de forskjellige grensesnittene og mobiltelefoner generelt. Vi hadde derimot noen hypoteser og påfølgende ambisjoner om hva vi ville undersøke. Det er ofte enklere å bekrefte en sikkerhetsrisiko enn å avkrefte. På grunn av dette var det vanskelig å "sette foten ned" når vi hadde søkt i flere timer på et emne. Risikoen var jo der for at vi ikke hadde lett på de rette plasser. Vi opplevde flere ganger at når vi fant svar på et spørsmål, ble vi møtt med flere nye spørsmål som måtte besvares. Dette førte til at vi brukte mye ressurser på noe som ikke fikk direkte relevans uten et lite skriv i vår loggbok. Vårt arbeid har også båret preg av å jobbe innimellom, utenom de ”vanlige” arbeidstidene. Vi har kanskje ved ren tilfeldighet kommet innom et emne som har vært relevant og forsket videre på dette. Det har derfor vært flere timer totalt som vi ikke har loggført.

Vi kunne vært flinkere til å bruke vår prosjektplan. Ved starten av prosjektet laget vi denne for å dekke hele prosjektperioden. Det tok derimot ikke lang tid før vi innså at vi måtte endre rekkefølge på flere aktiviteter. Igjen, på grunn av oppgavens omfang, var det vanskelig å estimere når det skulle arbeides med noe. Ikke minst var det vanskelig å estimere noe vi ikke visste om fantes. Vi har lagt ved original prosjektplan slik vi laget den ved prosjektstart, samt en oppdatert plan som viser hvordan arbeidet egentlig har blitt gjort. Sistnevnte er produsert på grunnlag av vår loggbok.

Utover dette er det flere områder som vi nå i ettertid gjerne skulle sett at vi hadde gjort litt anderledes. Disse punktene tar vi med oss videre.

- Sett at dette var en forskningsoppgave, kunne vi vært bedre til å strukturere våre søk, slik at vi i mindre grad gikk i gjennom samme resultat. Dette er ikke alltid like enkelt, spesielt ikke når vi bruker internett som vår primære informasjonskanal.
- Da vi brukte mye ressurser på å søke i blinde, burde vi hatt en prosess der vi kunne utelukket emner vi ikke fant svar på. Vi viser her spesielt til port 16650 som vi ikke fant ut hva var brukt til. Mye tid gikk ut på å ”prøve lykken” med forskjellige søke-

kombinasjoner. Når skal en sette foten ned å si at vi konkluderer med at svar på dette ikke er mulig å finne?

- Til tider ble vi så oppslukt i selve arbeidet at oppdatering av hjemmesiden gikk i glemmeboken. Ettersom dette bare tar fem minutter, burde vi ha klart å huske på dette.
- Vi hadde relativt få møter der alle parter var samlet. Dette hadde naturlige årsaker. Ettersom vi brukte en del tid på researching, ble det unaturlig å ha møter så ofte. Vi kunne ikke vise til så store resultater i starten og det ble derfor da mer praktisk å ha hyppigere dialoger per epost og telefon underveis.

Fordeling av arbeid

Fordeling av arbeid har stort sett gått greit. Vi har prøvd så langt som mulig å gjøre like mye. Vi har hatt interne møter når vi har hatt behov for det, der vi har diskutert status og videre arbeid. Hyppigheten av disse har vært alt fra en dag til et par dager. Dette har vi dog sett på som uformelle møter og har derfor ikke skrevet noe møteinkalling/møtereferat.

Vi har hatt en egen dedikert fil på vår repository (vårt lagringsområdet) hvor vi har samlet det vi har å gjøre av arbeidsoppgaver og andre saker vi ikke måtte glemme. Så lenge det har vært arbeidsoppgaver i denne fila, har gruppens medlemmer alltid hatt mulighet til å jobbe når enn det måtte være uten å måtte konferere med andre. Vi plukket som oftest ut en oppgave hver til det ikke flere mer igjen. Dette synes vi har fungert meget bra da vi har hatt muligheten til å jobbe med det vi har interessert oss mest for innenfor de gjeldene arbeidsoppgaver. I de tilfeller hvor arbeidsoppgavene kanskje ikke har vært like innbydende for gruppen, har gruppeleder fordelt de.

Prosjekt som arbeidsform

Prosjekt er en arbeidsform som virker å være/bli en populær arbeidsform i arbeidslivet. Det har derfor vært lærerikt å jobbe så tett med en større oppgave over lengre tid. Vi har fått erfart hvor lett det er å si/tenke at en oppgave tar kort tid, men at det i ettertid tok betraktelig lengre tid. Gruppen har også fått mer kjennskap til hvorfor prosjekter kan ende i fiasko. Vi har derimot klart å unngå dette.

Vedlegg C

Subjektiv opplevelse av bacheloroppgaven

Å jobbe med bacheloroppgaven har vært både krevende og interessant. Vi har lært mye spennende og har hatt en veldig bratt læringskurve da vi ikke hadde kunnskaper om noe av det vi har drevet med fra før av. Vi har fått erfaring med å jobbe tett i gruppe over lengre tid og hva dette har å si både på godt og ondt. Det har vært spesielt artig å se sluttrapporten begynne å ta form etter utallige tider med research. Det er svært moro at vår rapport vil bli brukt videre av vår oppdragsgiver for videreutvikling av sikkerhetsperspektivet rundt mobiltelefonbruk.

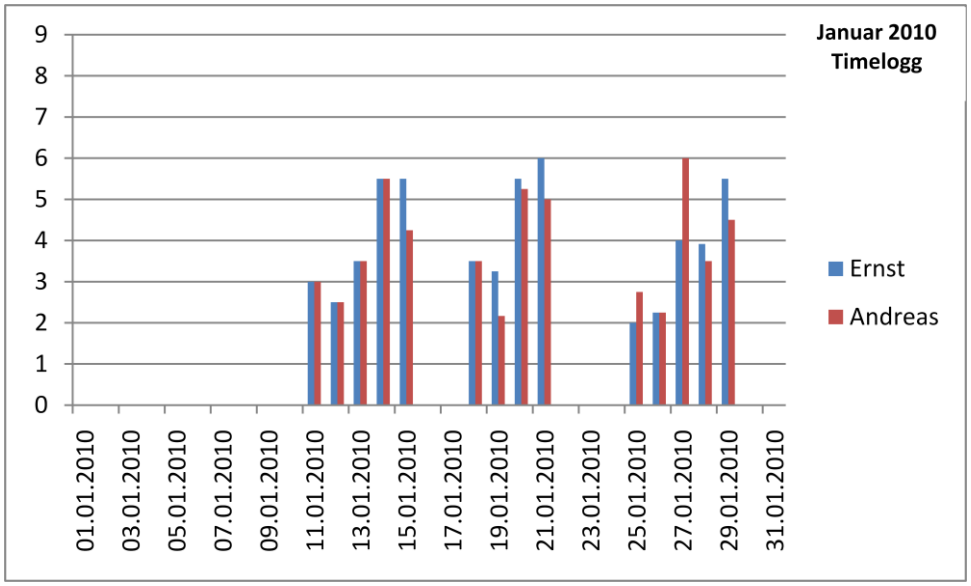
Fremgangsmåte / innhenting av informasjon.

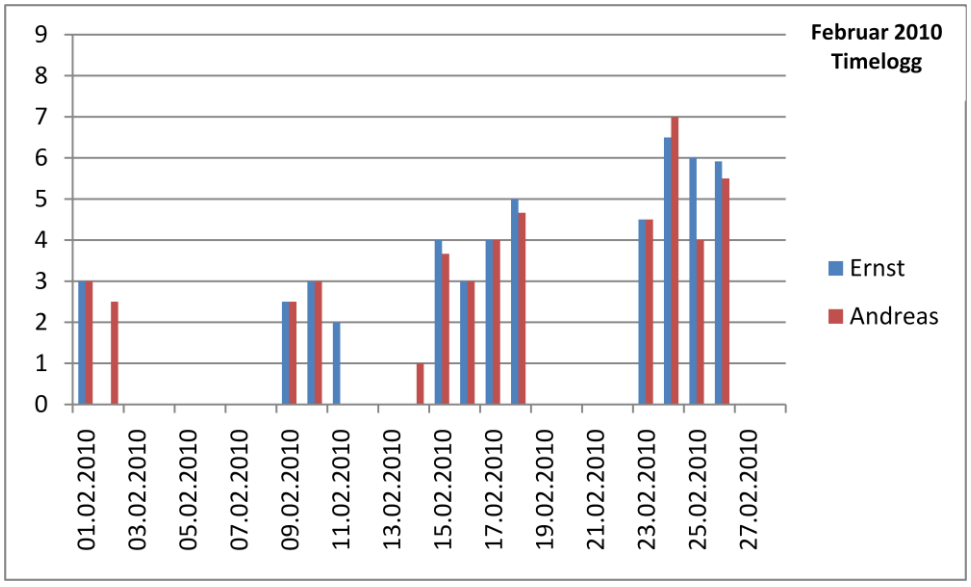
Sikkerhetsbildet er noe som forandrer seg hele tiden, noen ganger så ofte som fra dag til dag. Dette betyr at det ikke videre er praktisk å bruke bøker og lignende trykket litteratur som bakgrunn i en slik oppgave som denne, med mindre en dukker ned i selve standardiseringen av en teknologi. For eksempel Bluetooth. Vi har derimot ikke gjort dette, siden det i seg selv kunne vært en egen Bacheloroppgave. Vi valgte å se på mobilsikkerhet med en bredere vinkling for at vår oppdragsgiver skulle få så mye nyttig informasjon som mulig, og ut fra denne kunne danne seg et generelt bilde av om mobiltelefonen er sikker slik den er i dag. Vi valgte å fokusere på operativsystemet Android, og her finnes det ikke så veldig mye trykt materiale, i hvert fall ikke lærebøker. Majoriteten av våre kilder er derfor artikler og litteratur på internett, på grunn av de stadigere forandringene innen dette emnet.

Vedlegg D

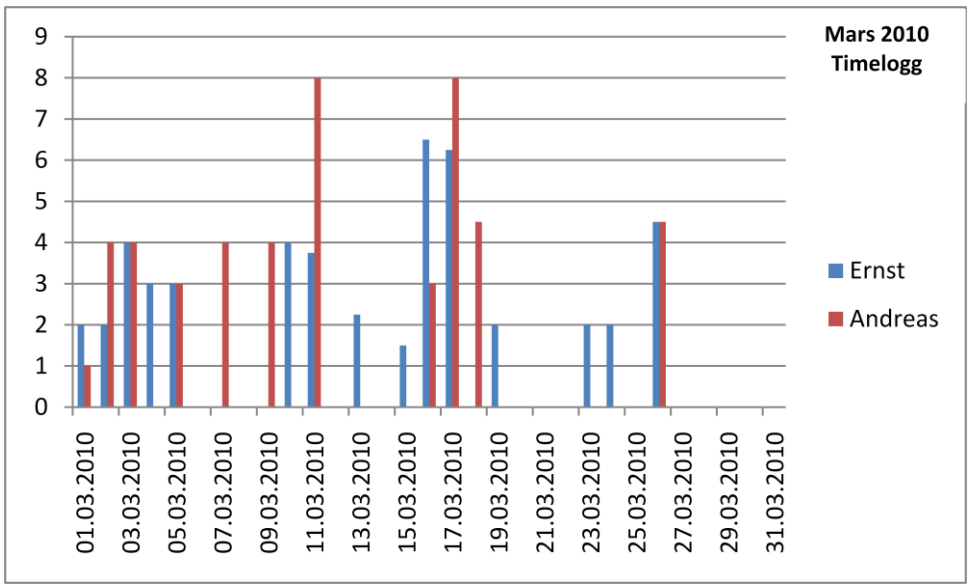
Timelogger

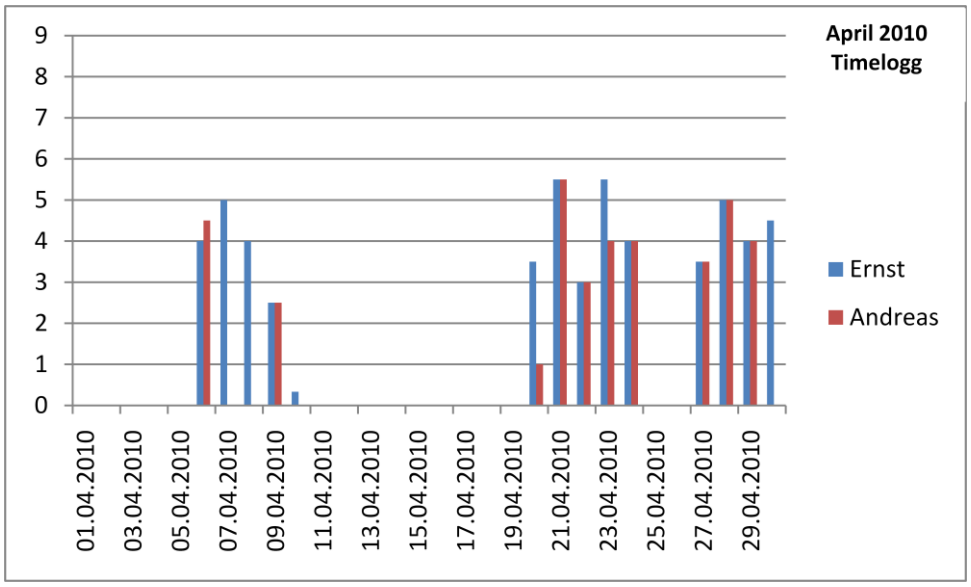
Under kan en se vår timelogg grafisk. Vi synes det er viktig å informere om at antall timer under er det minimum antall timer vi har arbeidet. Som nevnt før har det vært flere anledninger der vi ikke har fått notert ned vårt timeforbuk, eller fått notert ned riktig timeforbruk. Dette er noe vi burde ha fått til bedre.

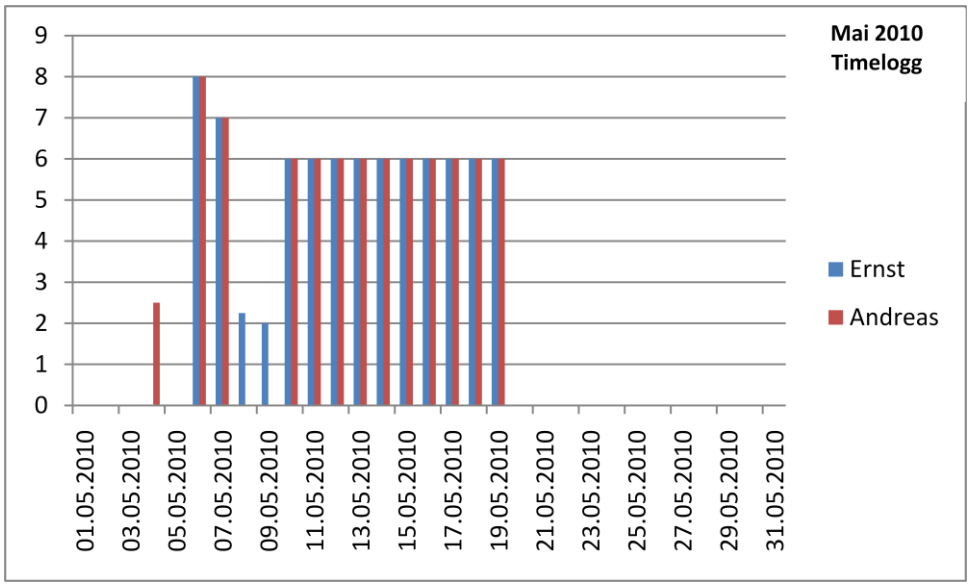




Vedlegg D


Mars 2010
Timelogg
Ernst
Andreas
0
1
2
3
4
5
6
7
8
9
01.03.2010
03.03.2010
05.03.2010
07.03.2010
09.03.2010
11.03.2010
13.03.2010
15.03.2010
17.03.2010
19.03.2010
21.03.2010
23.03.2010
25.03.2010
27.03.2010
29.03.2010
31.03.2010


April 2010
Timelogg
Ernst
Andreas
0
1
2
3
4
5
6
7
8
9
01.04.2010
03.04.2010
05.04.2010
07.04.2010
09.04.2010
11.04.2010
13.04.2010
15.04.2010
17.04.2010
19.04.2010
21.04.2010
23.04.2010
25.04.2010
27.04.2010
29.04.2010


Mai 2010
Timelogg
Ernst
Andreas
0
1
2
3
4
5
6
7
8
9
01.05.2010
03.05.2010
05.05.2010
07.05.2010
09.05.2010
11.05.2010
13.05.2010
15.05.2010
17.05.2010
19.05.2010
21.05.2010
23.05.2010
25.05.2010
27.05.2010
29.05.2010
31.05.2010

Vedlegg E

FORPROSJEKT:

Mobilsikkerhet

FORFATTER(E):

Ernst Kristian Henningsen (070992)

Andréas Bålsrud Berg (070347)

Dato: 22.01.10

Vedlegg E

Innhold

Intro

Etter vi gikk inn i informasjons-alderen har produksjonen av teknologiske enheter skutt i været. Datamaskiner spesielt, som før var forbeholdt de få personene som virkelig hadde bruk for det, er nå allemannseie. Det samme gjelder mobiltelefoner, før måtte en ha en sekk på ryggen for å transportere sin mobiltelefon fra A til B. Størrelsen på telefonene har blitt så små at dem nesten kan forsvinne i bukselomma, men dette til tross, har dem blitt mer og mer ressurssterke. De nyere mobiltelefonene i dag, bedre kjent som såkalte Smarttelefoner, kan gjøre stort sett de samme oppgavene som en svakere datamaskin kan.

Mobiltelefonen har med årene utviklet seg til å bli mye mer enn bare en telefon. Tjenester som tidligere bare var forbeholdt datamaskinen som nettsurfing, streaming, e-handel, nettbank etc., blir sakte men sikkert tilgjengelig på mobile enheter som er mye mer portable enn en datamaskin.

Malware og lignende som før har vært mest utbredt på datamaskiner, blir nå mer og mer utbredt på mobiltelefoner. Potensialet for å bli angrepet har så langt vært liten, men det har vært noen tilfeller. Noen peker på mangel av standarder og lite ressurser i enhetene som avgjørende for at mobiltelefonen har fått være i fred, men den tiden har forandret seg. Dette er en utvikling som en ikke må se vekk ifra. Det er stadig oftere oppslag i media om viruser, trojanere og phishing-forsøk som forekommer på mobiltelefoner.

Flere går rett og slett nå om dagen rundt med en liten datamaskin i lomma, uten å være klar over det. Ressurssterke mobiltelefoner, heretter kalt "Smarttelefoner", har kommet for å bli og tar flere og flere markedsandeler.

Disse Smarttelefonene sørger for at man alltid er i kontakt med omverdenen. Dette har sine positive og negative sider. De positive er selvfølgelig at det som en før gjorde på datamaskinen, nå kan gjøres mens en venter på bussen på holdeplassen. Denne fleksibiliteten som har blitt implementert i vår hverdag, kombinert med all den personlige informasjonen som flere brukere flittig lagrer på sin smarttelefon, utgjør også den negative siden. De er flere personer med uærlige hensikter som ser muligheten for å tjene penger på utnyttelse av denne teknologien, for eksempel ved å stjele bankkontodetaljer. Det er derfor veldig viktig å være bevisst på den potensielle risikoen en løper, når en bruker et slikt verktøy.

Vi vil med denne oppgaven derfor kartlegge sikkerhetsmekanismene i en av dagens typiske mobil-/smarttelefoner.

1. MÅL OG RAMMER

1.1 Vår bakgrunn

Gruppemedlemmene er studenter på Bachelor i Informasjonsikkerhet ved Høgskolen i Gjøvik. Vi har ingen tidligere kompetanse på telekommunikasjon eller mobiltelefoner, men valgte denne oppgaven fordi vi så på det som en fin utfordring hvor vi kan lære mye nyttig som kan anvendes senere i arbeidslivet. Oppgaven var en av flere forslag som ble foreslått til Høgskolen i Gjøvik. Vi har begge studert emnene "Ethical Hacking and Penetration Testing" og "Information Warfare" som vi tenker gir oss bedre kompetanse for dette prosjektet.

1.2 Prosjektmål

Effektmål:

Skape større fokus på mobiltelefonsikkerhet hos Sparebanken Hedmark og øke deres kompetanse på området.
Gjøre ansatte og kunder mer bevisste om risikoen ved bruk av smarttelefoner.

Resultatmål:

Produsere en rapport som gjør det enkelt å sette seg inn i de forskjellige sikkerhetsmekanismene og risikoene ved bruk av smarttelefoner, med spesiell vinkling mot operativsystemet Android.

Lage en demonstrasjon som tar for seg en eller flere sikkerhetstrusler mot smarttelefonen.

1.3Rammer

Testfase som inneholder eksperimentering med smarttelefoner, må gjøres i et isolert miljø.

Prosjektet skal være ferdig 20.05.10

2. OMFANG

2.1 Oppgavebeskrivelse

Etter oppdrag fra Sparebanken Hedmark, skal vi ta for oss Smarttelefonen, og undersøke hvilken risiko som kan foreløpe ved bruk på operativsystemet Android. Vi vil dykke ned og forklare hvilke sikkerhetsmekanismer som finnes i dette operativsystemet, og hvilken sikkerhet som ligger i de forskjellige protokollene og formatene for SMS og MMS samt se på hvordan disse blir behandlet etter de mottas av operativsystemet.
Vi vil også se på hvilken sikkerhet som ligger innenfor Bluetooth og WIFI og hvilke trusler disse kan ha ved bruk i en smarttelefon, samt hvilke muligheter det finnes for å kompromittere/angripe en mobiltelefon gjennom disse og hvordan man eventuelt kan skaffe seg uautorisert tilgang.
Ut ifra dette vil vi utvikle en detaljert rapport som vil kunne bli brukt av Sparebanken Hedmark for å forbedre deres kompetanse på området og skape fokus på mobilsikkerhet.
Vi vil utføre penetrasjonstesting og forsøke klientside-angrep på telefonen, samt forsøke å utvikle et SMS-virus/exploit.
Mot slutten av prosjektet vil vi utarbeide en presentasjon der vi vil demonstrere mulighetene en angriper har hvis han/hun oppnår tilgang til systemet, samt hvordan vi fikk tilgang hvis vi får til dette.

Våre problemstillinger er som følger:

- Hvilke sikkerhetsmekanismer foreligger I SMS, MMS, Bluetooth og WIFI relatert til operativsystemet Android? Hvilke muligheter finnes for å kompromittere/angripe en mobiltelefon gjennom disse?

- Hvilke tiltak kan innføres for å sikre disse og forhindre uautorisert tilgang og sikkerhetsbrudd på telefonen?

2.2 Avgrensning

2.2.1 Operativsystem

Ettersom operativsystemet står sentralt i smarttelefonene, diskuterte vi på vårt første møte med oppdragsgivere og veiledere, hvilke operativsystemer som kunne være aktuelle. Vi kom frem til at det stod mellom Windows Mobile, Android, Symbian og Iphone. Vi ble på dette møte også enige om at vi burde ta for oss et enkelt operativsystem, ettersom vi har begrenset med tid.
Sparebanken Hedmark stilte seg åpen for alle operativsystemene, det ble derfor opp til gruppa å velge.
Vi satte opp flere krav som operativsystemet måtte tilfredsstille. Kravene var som følger:

- Operativsystemet bør ha stor markedsandel/brukermasse, eller ha potensial til å oppnå dette innen et par år.

- Operativsystemet bør være lett tilgjengelig.

- Operativsystemet bør ha mulighet for å kjøre egenproduserte applikasjoner.

- Operativsystemet bør ha åpen kildekode, slik at sikkerhetsmekanismer kan studeres på lavere nivå hvis nødvendig.

Basert på dette kunne vi raskt luke vekk både Windows Mobile og Iphone, ettersom deres kildekode er lukket. Symbian sin kildekode var lukket helt frem til 2008 da Nokia tok over med sin visjon om å gjøre operativsystemet til åpen kildekode.[1] Vi sto derfor igjen med Symbian og Android (som også har åpen kildekode).

En stor forskjell på Symbian og Android er at Symbian allerede har en stor markedsandel, mens Android ikke har stor markedsandel per dags dato (januar 2010), ettersom det er ganske nytt. Ettersom vi har tro på at Android vil ta store markedsandeler de kommende årene endte valget på dette operativsystemet. Det var også avgjørende for oss at det ikke har blitt gjort grundig research på Android, ettersom det ikke har vært tilgjengelig mer enn noen få år.

2.2.2 GSM

Vi har blant annet lest gjennom kompendiet til vår veileder angående GSM.
På grunnlag av det som står i dette kompendiet, angående arkitektur og sikkerhetsarkitektur, har vi konkludert med at vi ikke vil dykke ned i/teste GSM.
Dette grunnet godt innarbeidet sikkerhet.

[1] http://www.symbian.org

2.2.3 USB

Vi har valgt å fokusere dette prosjektet på fjern-tilkobling. Vi vil derfor ikke utforske mulighetene ved tilgang direkte gjennom USB.

3. PROSJEKTORGANISERING

3.1. Ansvarsforhold og roller

Gruppen består av Andréas B. Berg og Ernst Kristian Henningsen.
Henningsen er gruppeleder.

Oppdragsgiver er Sparebanken Hedmark med Amund Mortensen og Thore Fjogstad som kontaktpersoner.

Akademiske veiledere ved HiG er Jan A. Audestad og Maciej Pietka.

3.2. Rutiner og regler i gruppa

Arbeidstid hver ukedag mellom 08:00 til 15:00, utenom mandag og tirsdag da vi har forelesning i annet fag. Disse dagene avtaler vi møtetidspunkt.
Hver fredag skal en kort statusrapport bli skrevet og bli lagt opp på vår prosjektnettside.
Hver 14.dag blir det holdt møte mellom gruppemedlemmer, veiledere og oppdragsgivere. Det vil bli sendt ut møteinnkalling i forkant. Møtene blir holdt annenhver gang på forholdsvis Høgskolen i Gjøvik og Sparebanken Hedmark sitt kontor på Hamar.

3.3 Arbeidsmetode

Gruppeleder delegerer litteratur som vi skal sette oss inn i. Hvert gruppemedlem er ansvarlig for å gå gjennom litteraturen til avtalt tid. Når vi møtes neste gang etter avtalt tid, går vi gjennom eventuelle merkverdige funn/kunnskap som det andre gruppemedlemmet bør vite. Disse funnene blir skrevet inn i et referat som vil inneholde alle funn – og i hvilket dokument/bok/etc. dem ble funnet. Dette for å gjøre det enkelt å lese mer om et spesifikt område på et senere tidspunkt.
Vi kommer utenom dette til å arbeide i vårt arbeidsrom ved Høgskolen i Gjøvik. Vi vil her gå gjennom litteratur som er lest, skrive rapporter og diskutere veien videre. Produksjon av rapporter og dokumenter vil foregå i fellesskap på vårt arbeidsrom i felleskap.

3.4 Ressursbehov

Vi trenger tilgang til Androids kildekode.

Vi trenger to smarttelefoner, en datamaskin og en switch/aksesspunkt til bruk i test-fasen. Dette for å kunne jobbe/teste i et isolert miljø.
Et arbeidsrom der vi kan oppbevare verdisaker som datamaskiner, smarttelefoner og dokumenter.

3.3.1 Budsjett

Møtekostnader
Vi må påregne oss en tur til Hamar ca en gang per måned for avholding av møter, vil vi her budsjettere med 150kroner tur-retur. Ettersom prosjektet vil gå over en fire-måneders periode, vil dette beløpe seg til 600kroner.
Ved møtene som blir holdt ved Høgskolen i Gjøvik må kostnader til kaffe og lignende beregnes. Vi regner 80kroner for dette per møte. Dette vil da totalt bli 320kroner.
Teknisk utstyr:
Vi har avtale med Sparebanken Hedmark om lån av to smarttelefoner med Android som vi kan bruke i forbindelse med testing.
Kontorkostnader:
Vi har betalt 300kroner i depositum for arbeidsrom. Dette vil bli tilbakebetalt til oss så lenge arbeidsrom blir forlatt i god tilstand.

Oppsummering
Vi budsjetterer at vi i løpet av denne prosjektperioden vil bruke 920kroner på prosjekt-relaterte kostnader. En potensiell økning av dette til 1220kroner på grunn av arbeidsrom er mulig, men lite sannsynlig.

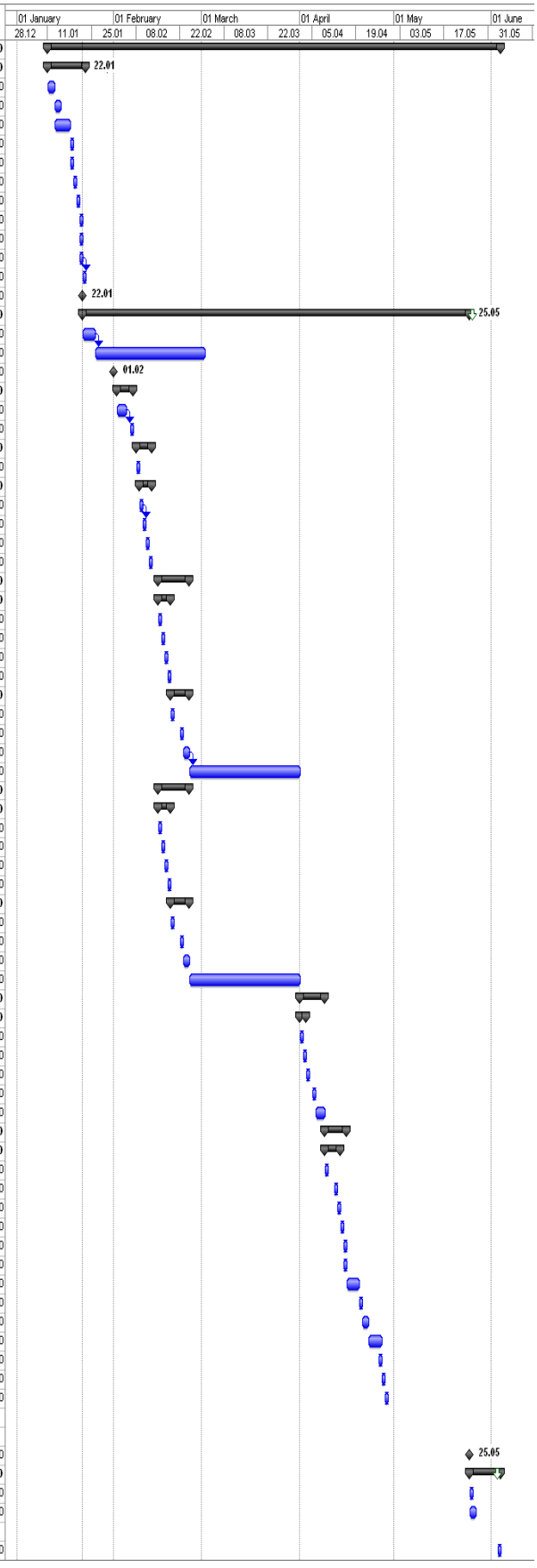
4. RISIKOER MED PROSJEKTET

Sannsynlighet: Skala 1 – 5
Konsekvens: Skala 1 – 5
Risiko: Skala 1 – 25
Der tallet 1 er lavest.

Risiko-område	Sannsynlighet	Konsekvens	Risiko	Tiltak
Finner ikke relevant informasjon om SMS/MMS, WIFI og Bluetooth	2	5	10	Start research så tidlig som mulig. Etablere forbindelser med ressurspersoner raskt.
Finner ikke (nok) teknisk informasjon om Android sine sikkerhetsmekanismer, samt hvordan de gjeldene grensesnittene/teknologiene blir håndtert.	2,5	5	12,5	Starte research så tidlig som mulig. Etablere forbindelser med ressurspersoner raskt.
Smarttelefoner ankommer oss sent i prosjektperioden. Dette vil medføre mindre tid til testing	2	5	10	Smarttelefoner er allerede bestilt. Vi bør holde oss oppdatert på status om leveranse. Hvis det går for lang tid før dem blir sendt/mottatt, må vi ta kontakt med leverandør/leveringstjenesten respektivt.
Exploit/virus for SMS/MMS blir ikke laget. Dette vil være et kutt i vår rapport som helhet	4	4	16	Starte med research på SMS/MMS så fort som mulig. Dette for å kunne sette seg inn i eventuelt nytt programmeringsspråk over tid.
Overskride budsjett	1	1	1	Tenke godt over økonomiske avgjørelser.
Presentasjon for kunder/ansatte ikke ferdig	1,5	5	7,5	Lage en god arbeidsplan

5. Gannt-skjema

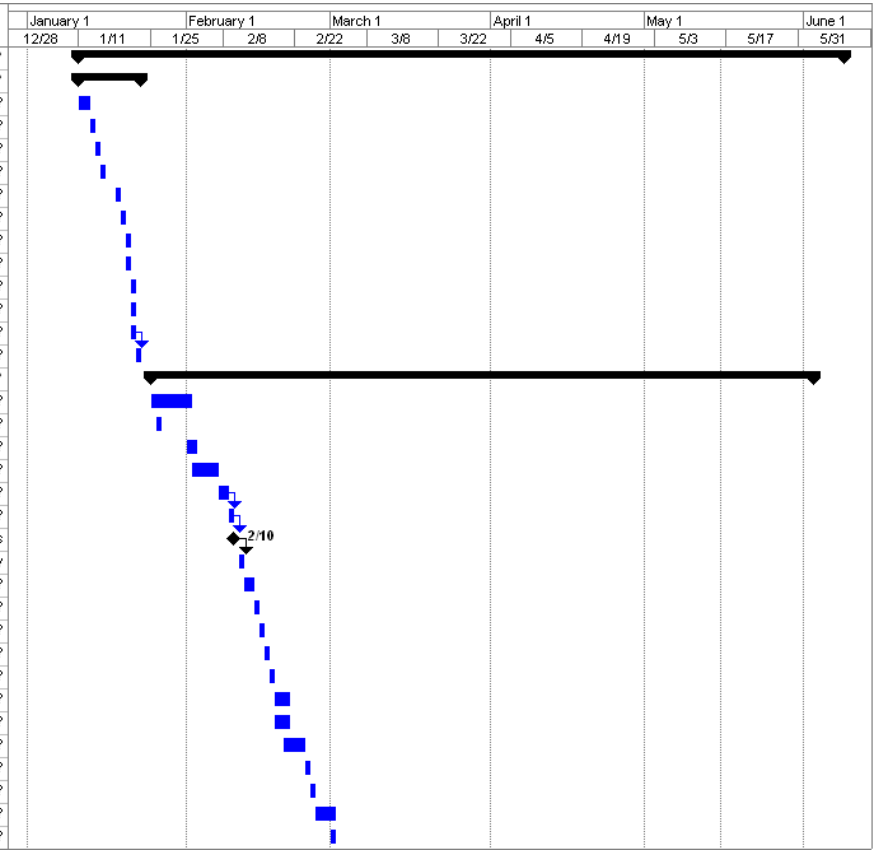
ID	Task Name	Duration	Start	Finish
1	**Prosjekt Mobilsikkerhet**	**106 days?**	**Mon 11.01.10**	**Thu 03.06.10**
2	**Utvikle forprosjekt**	**10 days?**	**Mon 11.01.10**	**Fri 22.01.10**
3	Diverse oppsetting/administrativt arbeide	2 days?	Mon 11.01.10	Tue 12.01.10
4	Brainstorming, hvilke vinklinger kan være aktuelle?	2 days?	Wed 13.01.10	Thu 14.01.10
5	Researche på vinklinger	3 days?	Wed 13.01.10	Sun 17.01.10
6	Internt møte i gruppa om resultat fra research	1 day?	Mon 18.01.10	Mon 18.01.10
7	Begrensning av vinklinger	1 day?	Mon 18.01.10	Mon 18.01.10
8	Formulere problemstillinger	1 day?	Tue 19.01.10	Tue 19.01.10
9	Formulere prosjektbeskrivelse	1 day?	Wed 20.01.10	Wed 20.01.10
10	Sette sammen produsert informasjon til forprosjekt rev1	1 day?	Thu 21.01.10	Thu 21.01.10
11	Sende forprosjekt rev1 til veileder	1 day?	Thu 21.01.10	Thu 21.01.10
12	Endre forprosjekt rev1 på bakgrunn av kommentarer	1 day?	Thu 21.01.10	Thu 21.01.10
13	Ferdigstille forprosjekt	1 day?	Fri 22.01.10	Fri 22.01.10
14	Levere ferdig forprosjekt til veileder	0 days	Fri 22.01.10	Fri 22.01.10
15	**Hovedprosjekt**	**89 days?**	**Fri 22.01.10**	**Tue 25.05.10**
16	Finne ut og innhente litteratur vi trenger	2 days?	Fri 22.01.10	Mon 25.01.10
17	Sette oss inn i gjeldende litteratur	26 days?	Tue 26.01.10	Mon 01.03.10
18	Møte med oppdragsgiver og veileder	0 days	Mon 01.02.10	Mon 01.02.10
19	**Researche/skrive intro om GSM**	**5 days?**	**Tue 02.02.10**	**Sat 06.02.10**
20	Hvilke sikkerhetsmekanismer finnes i GSM	3 days?	Tue 02.02.10	Thu 04.02.10
21	Konklusjon om GSM	1 day?	Sat 06.02.10	Sat 06.02.10
22	**Researche Android**	**5 days?**	**Mon 08.02.10**	**Fri 12.02.10**
23	Innhente kildekode til Android	1 day?	Mon 08.02.10	Mon 08.02.10
24	**Hvilke sikkerhetsmekanismer eksisterer i Android**	**4 days?**	**Tue 09.02.10**	**Fri 12.02.10**
25	Offentlig fra Google	1 day?	Tue 09.02.10	Tue 09.02.10
26	Er dem sikkre?	1 day?	Wed 10.02.10	Wed 10.02.10
27	Pakker som er kjent for dårlig sikkerhet	1 day?	Thu 11.02.10	Thu 11.02.10
28	Konklusjon om sikkerhetsmekanismer	1 day?	Fri 12.02.10	Fri 12.02.10
29	**Researche SMS**	**8 days?**	**Mon 15.02.10**	**Wed 24.02.10**
30	**Utgreiing**	**4 days?**	**Mon 15.02.10**	**Thu 18.02.10**
31	Protokoll	1 day?	Mon 15.02.10	Mon 15.02.10
32	Format	1 day?	Tue 16.02.10	Tue 16.02.10
33	Mulighet for å endre en SMS-pakke	1 day?	Wed 17.02.10	Wed 17.02.10
34	Sikkerhetsmekanismer, hva hindrer eventuelt kjøring av	1 day?	Thu 18.02.10	Thu 18.02.10
35	**Hvordan håndterer Android SMS?**	**4 days?**	**Fri 19.02.10**	**Wed 24.02.10**
36	Utgreiing	1 day?	Fri 19.02.10	Fri 19.02.10
37	Sikkerhetsmekanismer	1 day?	Mon 22.02.10	Mon 22.02.10
38	Mulige exploits/virus, eksekvering av kode gjennom SMS	2 days?	Tue 23.02.10	Wed 24.02.10
39	Skrive et virus/exploit - SMS	25 days?	Thu 25.02.10	Wed 31.03.10
40	**Researche MMS**	**8 days?**	**Mon 15.02.10**	**Wed 24.02.10**
41	**Utgreiing**	**4 days?**	**Mon 15.02.10**	**Thu 18.02.10**
42	Protokoll	1 day?	Mon 15.02.10	Mon 15.02.10
43	Format	1 day?	Tue 16.02.10	Tue 16.02.10
44	Mulighet for å endre en MMS-pakke	1 day?	Wed 17.02.10	Wed 17.02.10
45	Sikkerhetsmekanismer, hva hindrer eventuelt kjøring av	1 day?	Thu 18.02.10	Thu 18.02.10
46	**Hvordan håndterer Android MMS?**	**4 days?**	**Fri 19.02.10**	**Wed 24.02.10**
47	Utgreiing	1 day?	Fri 19.02.10	Fri 19.02.10
48	Sikkerhetsmekanismer	1 day?	Mon 22.02.10	Mon 22.02.10
49	Mulige exploits/virus, eksekvering av kode gjennom MMS	2 days?	Tue 23.02.10	Wed 24.02.10
50	Skrive et virus/exploit - MMS	25 days?	Thu 25.02.10	Wed 31.03.10
51	**Researche WIFI**	**7 days?**	**Thu 01.04.10**	**Thu 08.04.10**
52	**Utgreiing**	**2 days?**	**Thu 01.04.10**	**Fri 02.04.10**
53	Protokoller	1 day?	Thu 01.04.10	Thu 01.04.10
54	Kryptering	1 day?	Fri 02.04.10	Fri 02.04.10
55	Man in the middle angrep	1 day?	Sat 03.04.10	Sat 03.04.10
56	Sniffing	1 day?	Mon 05.04.10	Mon 05.04.10
57	Penetrasjonstest - Nmap++ av testtelefon	3 days?	Tue 06.04.10	Thu 08.04.10
58	**Researche Bluetooth**	**5 days?**	**Fri 09.04.10**	**Thu 15.04.10**
59	**Utgreiing**	**3 days?**	**Fri 09.04.10**	**Tue 13.04.10**
60	Protokoll	1 day?	Fri 09.04.10	Fri 09.04.10
61	Format	1 day?	Mon 12.04.10	Mon 12.04.10
62	Sikkerhetsmekanismer	1 day?	Tue 13.04.10	Tue 13.04.10
63	Mulighet for å endre en Bluetooth pakke	1 day?	Wed 14.04.10	Wed 14.04.10
64	Kjente metoder for angrep	1 day?	Thu 15.04.10	Thu 15.04.10
65	Eventuelt: Testing av angrep	1 day?	Thu 15.04.10	Thu 15.04.10
66	Sette sammen alle dokument	2 days?	Fri 16.04.10	Mon 19.04.10
67	Lese gjennom samensatt dokument (rød tråd)	1 day?	Tue 20.04.10	Tue 20.04.10
68	Skrive konklusjon	2 days?	Wed 21.04.10	Thu 22.04.10
69	Lese gjennom sluttrapport-første utkast, noe vi mangler?	2 days?	Fri 23.04.10	Mon 26.04.10
70	Endre sluttrapport-første utkast	1 day?	Mon 26.04.10	Mon 26.04.10
71	Lese gjennom sluttrapport-andre utkast (Noe vi mangler? Korrektu	1 day?	Tue 27.04.10	Tue 27.04.10
72	Endre sluttrapport-andre utkast	1 day?	Wed 28.04.10	Wed 28.04.10
73				
74				
75	Innlevering av Prosjekt til studenttorget	0 days	Tue 25.05.10	Tue 25.05.10
76	**Arbeide med presentasjon**	**8 days?**	**Tue 25.05.10**	**Thu 03.06.10**
77	Finne foiler som skal brukes (fargevalg etc.)	1 day?	Tue 25.05.10	Tue 25.05.10
78	Trekke ut hvilken informasjon vi vil gi til publikum	2 days?	Tue 25.05.10	Wed 26.05.10
79				
80	Presentasjon av Prosjekt, ved HIG	1 day?	Thu 03.06.10	Thu 03.06.10



Faktisk Gannt-skjema

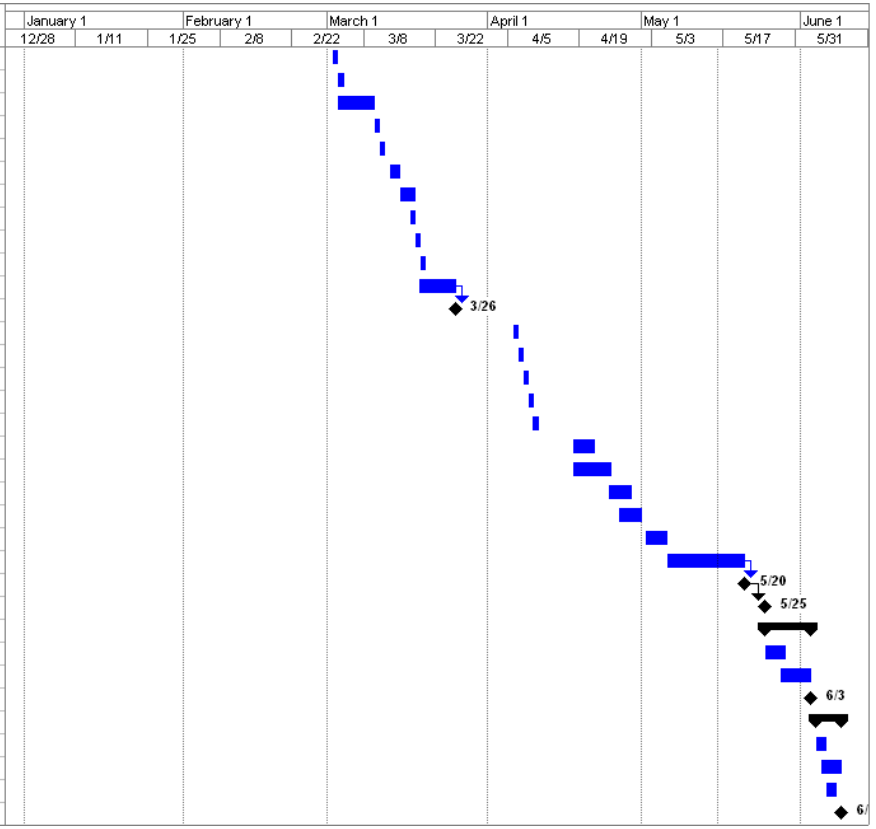
Som nevnt, synes vi selv at vi kunne vært en god del bedre på å bruke en slik plan. En lager ikke slike planer fordi en må, en lager de for de skal være et hjelpemiddel for å nå målene en ønsker. Ettersom vi opplevde at det var vanskelig å planlegge langt frem i tid, fikk ikke vi utbytte av vår plan i den grad vi ønsket. Vi ser derimot klare fordeler ved å bruke en slik plan mer aktivt, og det er noe vi tar til oss på veien videre etter vår Bachelor-grad, om det så er i Mastergrad- eller jobb-samenheng. Under har vi satt sammen en plan som illustrerer når vi arbeidet med de forskjellige emnene. (Planen er fordelt på denne og neste side). Som en kan lese ut fra planen, har det vært vanskelig å gjøre seg ferdig med et emne, ettersom det ofte dukket opp flere vinklinger senere. Planen er laget på bakgrunn av vår prosjektdagbok, der vi noterte dato for arbeid, hva vi gjorde, og antall timer.

ID	Task Name	Duration
1	**Prosjekt Mobilsikkerhet**	**149 days?**
2	**Utvikle forprosjektrapport**	**12 days?**
3	Diverse oppsetting/administrativt arbeide	2 days?
4	Brainstorming, avgrenset prosjekt	1 day?
5	Skrevet grupperegler, startet med utarbeiding av oppgaven	1 day?
6	Utarbeiding av oppgaven, skrevet I forprosjekt	1 day?
7	Utarbeidet aktivitetsliste, vurdert arbeidsmetodikk	1 day?
8	Dialog med oppdragsgiver angående problemstillinger	1 day?
9	Evaluert risikoer med oppgaven, skrevet I forprosjektrapport	1 day?
10	Sendt forprosjektrapport til veileder	1 day?
11	Prat med veileder, gitt oss klarsignal på prosjekt	1 day?
12	Lest om SMS og malware	1 day?
13	Ferdigstillt forprosjekt	1 day?
14	Sendt ferdig forprosjekt til veileder	1 day?
15	**Hovedprosjekt**	**129 days?**
16	Funnet og strukturert litteratur om Android, skrevet om Android	8 days?
17	Sendt epost til Norsis og NPT (kontantkortabonnement og malware)	1 day?
18	Bitt kjent med Android-emulator	2 days?
19	Funnet relevant litteratur og skrevet om malware/overvåkning av telefon	5 days?
20	Gått gjennom hva vi har funnet til nå	2 days?
21	Laget og sendt ut agenda for prosjektmøte	1 day?
22	Prosjektmøte	0 days
23	Gått gjennom møtereferat og hva vi skal fokusere på videre	1 day
24	Innhenting av informasjon, sendt ut eposter	2 days?
25	Forskjell på Android firmware 1.5 og 2.1	1 day?
26	Kontaktet Google, oppdatert nettside	1 day?
27	Innhentet informasjon om rooting, forskjeller på versjon 1.5 og 2.1	1 day?
28	Sendt epost om kontantkortabonnement, rooting, forskjell på 1.5 og 2.1	1 day?
29	Skrevet om kontantkortabonnement	3 days?
30	Skrevet om Malware	3 days?
31	Skrevet/strukturert hoveddokument, malware	4.13 days?
32	Sett film angående WARSUCKING, sårbarhetstesting, dialog med JAN, rooting	1 day?
33	Bluetooth, verktøy og muligheter. MMS, ADB, filsystem	1 day?
34	Sendt ut eposter, skrevet I hoveddokument, SMS-applikasjon	4 days?
35	Bluetooth testing, researching, IRC-ressurs	1 day?



Vedlegg F

ID	Task Name	Duration
36	Bluetooth testing, researching, java, Android SDK	1 day?
37	Bluetooth, WIFI, SMS-applikasjon	1 day?
38	WIFI - Portscanning og research, eclipse, java, SMS-applikasjon	7 days?
39	Dialog med Slobodan (HIG) om Bluetooth, satt oss inn I dette	1 day?
40	Testing av DimDim, sendte eposter, bluetooth, skype, android java app	1 day?
41	Ryddet/sortert vårt lagringsområde	2 days?
42	Skrevet om portscan, programmering	3 days?
43	Startpakke, lynkurs om rapport, programmering	1 day?
44	Skrevet om programmering	1 day?
45	Skrevet om kontantkortabonnement	1 day?
46	Forberedelse til møte	7.13 days?
47	Prosjektmøte	0 days
48	Gått gjennom referat fra møte, krav til leverandør, funksjoner på telefon, brosjyre	1 day?
49	MMS, dialog med veileder, gannt, dialog med medie-foreleser ang tegneserie, sendt epost	1 day?
50	MMS, ideer til brosjyre, hva har vi funnet ut til nå?	1 day?
51	Internt møte om risiko-vurdering, skrevet og strukturert hoveddokument, Dalvik VM	1 day?
52	Webkit	1.13 days?
53	Internt møte om status, rapport-fokus	4.13 days?
54	Internt møte, brosjyre, port 16650	7.13 days?
55	Brosjyre	4.13 days?
56	Internt møte, rapport, bluetooth, brosjyre	4.13 days?
57	Brosjyre	4.13 days?
58	Rapport	15 days?
59	Innlevering av rapport til trykking	0 days
60	Innlevering av trykkt rapport til studenttorget	0 days
61	**Arbeide med presentasjon ved Høgskolen I Gjovik**	**9 days?**
62	Trekke ut hvilken informasjon vi vil gi til publikum	4 days?
63	Finne foiler som skal brukes (fargevalg etc.)	6 days?
64	Presentasjon av Prosjekt, ved HIG	0 days
65	**Arbeide med presentasjon for Sparebanken Hedmark**	**5 days?**
66	Finne ut hva vi skal presentere, I forhold til publikum	2 days?
67	Forberede demonstrasjon av SMS-app + evt mobistealth	4 days?
68	Sette informasjon inn I presentasjonsmateriell	2 days?
69	Presentasjon av rapport, for Sparebanken Hedmark	0 days



Risikovurdering av prosjektet, per 17.02.2010

Risiko-område	Sannsynlighet	Konsekvens	Risiko	Tiltak
Finner ikke relevant informasjon om SMS/MMS, WIFI og Bluetooth	2	5	10	Start research så tidlig som mulig. Etablere forbindelser med ressurspersoner raskt.
Finner ikke (nok) teknisk informasjon om Android sine sikkerhetsmekanismer, samt hvordan de gjeldene grensesnittene/teknologiene blir håndtert.	3,5	5	17,5	Etablere kontakt med ressurspersoner raskt. Dreie oppgaven mot det vi har informasjon om
Exploit/virus for SMS/MMS blir ikke laget. Dette vil være et kutt i vår rapport som helhet.	4	4	16	Lete etter eksisterende arbeid om dette. Kombinere flere angrepsvektorer. Vurdere muligheten for å droppe virus-laging og heller fokusere mer på hands-on testing av telefonene. Vi må vite fort om det skal være aktuelt ettersom vi antakeligvis må sette oss inn i nytt programmeringsspråk.
Overskride budsjett	1	1	1	Tenke godt over økonomiske avgjørelser.
Presentasjon for kunder/ansatte ikke ferdig	1,5	5	7,5	Lage en god arbeidsplan. Opprettholde dialog med Sparebanken Hedmark om hva dem ønsker i presentasjonen.
Rooting av telefoner går skeis	2	5	10	Ta grundig backup av ROM før rooting.

Vanskelig å holde status på fremdrift, kan føre til tidsproblem i slutt av prosjektet. Vanskelig å finne rett informasjon til rett tid. Blir mye hopping mellom emner. Vanskelig å gjøre seg ferdig med et gitt emne ettersom en finner ny informasjon hele tiden.	**4**	**5**	**20**	**Akseptere at vi må arbeide med flere emner parallelt. Skrive ned nye fakta ettersom vi får dem, med kildehenvisninger.**
Blir sittende med for mye informasjon, vanskelig å velge hvilken informasjon som bør vektlegges	**3**	**3**	**9**	**Diskutere hva vi vil presentere for Sparebanken Hedmark, og andre som leser rapporten. Hva er interessant?**
Baserer informasjon på usikre kilder				**Vurdere de forskjellige forfatterne, om de har gitt ut andre artikler**
Brosjyre blir ikke ferdig i tide				

Risikovurdering av prosjektet, per 09.04.2010

Risiko-område	Sannsynlighet	Konsekvens	Risiko	Tiltak
Finner ikke relevant informasjon om Dalvik VM / Sandbox-miljøet	3	3	9	Søke aktivt på nettet. Ta eventuelt kontakt med brukere på IRC-kanalen
Finner ikke ut noe om prosedyrer ved mobilbank	3	3	9	Når vi får mail fra skandiabanken må vi raskt evaluere om vi skal sende ut flere mailer.
Presentasjon for ansatte/skolen ikke ferdig	1,5	5	7,5	Lage en god arbeidsplan. Opprettholde dialog med Sparebanken Hedmark om hva dem ønsker i presentasjonen. Idemyldring av hva vi kan snakke om. Skremselspropaganda
Vanskelig å holde status på fremdrift, kan føre til tidsproblem i slutt av prosjektet. Vanskelig å finne rett informasjon til rett tid. Blir mye hopping mellom emner. Vanskelig å gjøre seg ferdig med et gitt emne ettersom en finner ny informasjon hele tiden.	4	5	20	Akseptere at vi må arbeide med flere emner parallelt. Skrive ned nye fakta ettersom vi får dem, med kildehenvisninger.
Baserer informasjon på usikre kilder	4	3	12	Vurdere de forskjellige forfatterne, om de har gitt ut andre artikler. Vurdere om informasjon som foreligger virker logisk. Sunn fornuft.
Brosjyre blir ikke ferdig i tide	3	5	15	Finne ut raskt hva vi skal

				ha i brosjyre, snakke med Sparebanken. Opprette kontakt med utenforstående personer raskt (tegnere og lignende).
Rapporten blir ikke ferdig i tide	**1,5**	**5**	**7,5**	**Jobbe effektivt med de arbeidsoppgaver vi vet om. Skrive mye på rapporten fremover. Bruke hverandre til å se over det som er skrevet.**

Vedlegg H.1

Møteinnkalling

Dato: 07.Januar 2010
Klokkeslett 12:00 – 15:00
Møtested: Høgskolen I Gjøvik, Ørneredet?

Deltakende
Andréas B. Berg
Ernst Kristian Henningsen
Maciej Pietka
Jan A, Audestad
Amund Mortensen
Thore Fjogstad

Agenda

Diskusjon
-Hva vil Sparebanken Hedmark oppnå gjennom denne oppgaven?

Retning/begrensning av oppgaven
-Informasjonsflyt (GSM/3G/EDGE/WIFI/BLUETOOTH etc)
-Systemarkitektur
-Sikkerhetsarkitektur
-Grensesnitt (simulering/fysiske enheter)

Hvilke oppgaver bør det fokuseres på?
(Reformulere prosjektplan)
(Fremtidige møter)

Vedlegg H.2

Møtereferat

Andréas B. Berg (referent)

Dette var et møte angående oppstart av hovedprosjektet 2010-01-07.

Disse var med på møtet: Andréas B. Berg
Ernst Kristian Henningsen
Jan A. Audestad
Amund Mortensen
Thore Fjogstad

Disse var fraværende: Maciej Pietka

Møteleder var: Ernst Kristian Henningsen

På dette møtet ble det diskutert hva Sparebanken Hedmark ønsker å oppnå med denne oppgaven og hvilke rammer vi burde forholde oss til med tanke på teknologi, muligheter og tidligere kompetanse.

Det ble bestemt at vi (Ernst og Andréas) skal bestemme oss for hvilke temaer og områder av mobilsikkerhet vi vil ha med i oppgaven og hvilke dypdykk vi ønsker å ta. Vi skal også bestemme oss for hvilket operativsystem vi ønsker å jobbe med, og gjøre research på de emnene vi bestemmer oss for samt å utarbeide en forprosjektplan.

Neste møte er ikke enda fastsatt, men vil bli planlagt og bestemt over mail og telefon.

Vedlegg H.3

Møteinnkalling

Dato: 10.02.2010
Klokkeslett: 13:00 – 15:00(?)
Møtested: Sparebanken Hedmark, Torggt. 12-14, 2302 Hamar

Deltakende
Andréas B. Berg
Ernst Kristian Henningsen
Maciej Pietka
Amund Mortensen
Thore Fjogstad

Agenda

- Kort introduksjon av Android, oppbygging

- Muligheter med SMS
  - Spoofing
  - Sporbarhet

- Android Marked

- Finne nye exploits eller se videre på eksisterende?

- Testing av telefoner (testmiljø)

- Veien videre..

- Annet?

Ta kontakt på mail hvis det er noen spørsmål.
Ernst kan ellers nås på telefon 476 42 235
Andreas kan nås på telefon 90 99 66 91

Vedlegg H.4

Møtereferat

Andréas B. Berg (referent)

Dette var et møte angående status og videre arbeid, 10.02.2010

Disse var på møtet: Andréas B. Berg
Ernst Kristian Henningsen
Maciej Pietka
Amund Mortensen
Thore Fjogstad

Disse var fraværende: Jan A. Audestad

Møteleder var: Ernst Kristian Henningsen

På møtet ble det presentert hvilken informasjon vi har funnet hittil og hvilke tanker vi har rundt oppgaven videre. De forskjellige emnene ble diskutert for å være sikre på at partene er oppdatert i prosjektet.

Vi, Ernst og Andréas, fant flere punkter vi må finne mer informasjon om, samt noen enkelte nye punkter. Vi vil nå fremover fokusere på dette, samt kontakte NorSIS, Google Norway og et par teleoperatører for å tilegne oss mer informasjon.

Vi fikk på dette møtet utlånt de to telefonene vi skal foreta tester på:
- HTC HERO (1.5)
- NEXUS ONE (2.1)

Neste møte er ikke fastsatt, men vil sannsynligvis foregå over konferansetelefon og/eller DimDim. Tidspunkt vil bli planlagt og bestemt over mail og telefon.

Vedlegg H.5

Møteinnkalling

Dato: 26.03.2010
Klokkeslett: 11:15 – 13:00
Møtested: Telefonkonferanse

Deltakende
Andréas B. Berg
Ernst Kristian Henningsen
Maciej Pietka
Jan Arild Audestad
Amund Mortensen
Thore Fjogstad

Agenda

- Status på fremdrift, og hva vi har funnet ut til nå

- Hva skal vi fokusere på fremover

- Annet?

Praktisk informasjon
Vi er tre parter i telefonkonferansen
Sparebanken Hedmark
Gruppens medlemmer + Maciej Pietka
Jan Arild Audestad

I tillegg vil vi bruke presentasjonsverktøyet Dimdim (går gjennom browseren). Dere vil få tilsendt en mail med invitasjon. Når møte nærmer seg går dere inn på lenken som nevnt i mailen. Dette verktøyet vil bli brukt til å vise powerpointpresentasjonen.

Ta kontakt på mail hvis det er noen spørsmål.
Ernst kan ellers nås på telefon 476 42 235
Andreas kan nås på telefon 90 99 66 91

Vedlegg H.6

Møtereferat

Ernst Kristian Henningsen (referent)

Dette var et møte angående status og videre arbeid, 26.03.2010
Møtet tok sted over telefonkonferanse, der presentasjonsverktøyet Dimdim ble brukt for å vise en powerpoint-presentasjon

Disse var på møtet: Andréas B. Berg
Ernst Kristian Henningsen
Maciej Pietka
Jan A. Audestad
Amund Mortensen
Thore Fjogstad

Møteleder var: Ernst Kristian Henningsen

På møtet ble det presentert hvilken informasjon vi har funnet hittil og hvilke tanker vi har rundt oppgaven videre. Gruppen gjorde en powerpoint-presentasjon av hva som er funnet ut, denne ble vist til alle oppmøtte.

Vi ble enige om å sette presentasjonsdato ved Sparebanken Hedmark (for ansatte/kunder) til den **8.Juni 2010 klokken 09:00**

Det ble diskutert offentliggjøring av rapport, noe som gruppens medlemmer stiller seg positive til. Vi (Andreas og Ernst) ønsker at de som kan ha nytte av vår rapport, får tilgang til den, med en betingelse om at de refererer til oss, noe som sådan stort sett er vanlig i slike samenhenger.

Neste møte er ikke fastsatt, men vil ta sted kort tid etter påskeferien 2010. Tidspunkt vil bli planlagt og bestemt over mail/telefon. Vi ønsker prosjektgruppen en god påskeferie.

Under, på neste side, er emner som ble nevnt på møte (og noe relevant utenfor møte).

Konkrete emner som ble tatt frem på møte, som kan være interessante å gå nærmere inn på, i tillegg til tanker vi allerede har gjort rundt dette:

Se mer inn på muligheter ved bluetooth

Hvordan blir PINkode beskyttet, når telefonen er rootet?

Lage en brosjyre for Sparebanken Hedmark
- Hva skal denne inneholde av informasjon? (Konseptet)
- Punkter
- Målgruppe

MMS

Mobilbank

Brukere av telefoner får ofte flere tjenester som de selv aldri kommer til å bruke, dette kan være en sikkerhetsrisiko:
- Hva bør leverandører bruke/gjøre for at det skal bli sikrere å bruke en smarttelefon i dag?
- Må det settes krav til leverandør? Innføre en standard?
- Selger av en smarttelefon og kunde kan ha en dialog om hvilke tjenester som kunden trenger. Telefonen blir skryddersydd til dette slik at tjenester ikke blir overflødige. (I dag er det slik at telefonen er ferdig konfigurert ved kjøp).

Funksjoner til en smarttelefon og medfølgende sikkerhetsrisiko, hvis skadelig programvare er innstallert på telefon kan for eksempel følgende inntreffe:
- Kamera: Aktiveres og brukes for å spionere på konfidensiell informasjon.
- Lydopptaker: Aktiveres og brukes for å spionere/ta opp en konfidensiell samtale
- Bluetooth: Aktiveres og brukes for å overføre informasjon til en kriminell ved for eksempel utgang av arbeidssted.

Overstående (og alle andre funksjoner) klassifiseres etter sikkerhetsrisiko, slik at en enkelt kan ta en diskusjon om en må deaktivere de ”alvorligste tilfellene”.

”I hvor stor grad kan aktuell telefon utnyttes til uetiske formål?”

Smarttelefoner er banebrytende i den forstand at en nå kan ha en datamaskin i lomma. Derfor følger flere av de sikkerhetsrisikoene som eksisterer i ”dataverdenen”. Nedlasting av applikasjoner fra Android Market kan likestilles med å laste ned programvare fra en hvilken som helst side på nettet. Ved å deaktivere Android Market vil en derfor kunne hindre mesteparting av spredning med tanke på skadelig programvare. Derimot kan det tenkes at dette må deaktiveres slik at en ikke kan aktivere når en ønsker, for det er jo ikke Android Market i seg selv som er det skadelige, det er brukeren som laster ned fra dette markedet. I en bedrift-samenheng, der telefonen ønsker å ta i bruk de ressursene en smarttelefon kan tilby,

kunne det vært aktuelt at telefoner blir gitt ut av for eksempel IT-avdeling. Hver ansatt har en dialog med avdelingen om hvilke tjenester som trengs. Avdelingen gjør alle konfigureringer slik at telefon er tilpasset den enkelte ansatte, men ifølge bedriftens retningslinjer. Dette innebærer derimot at IT-avdelingen må være oppdatert på hvilken programvare som er sikker.

En konklusjon i vår rapport vil trolig omhandle tiltak som bør iverksettes for at sikkerheten skal bedres. Det kan blant annet være at Android Market BØR innføre sjekker på opplastete applikasjoner. Dette vil nok ikke garantere at markedet blir fri for skadelig programvare, men kunne luke vekk mye, og derav minske risikoen for å bli infisert. Våre nevnte tiltak vil antakeligvis være anbefalinger og ikke noe som vil være gjennomførbart rent praksis innenfor vår tidsramme i dette prosjektet.

Vi laget en ”proof-of-concept”-applikasjon som videresender alle mottatte meldinger til et valgt telefonnummer. Er det mulig å lage et annet proof-of-concept som åpner en bakdør på telefonen? For eksempel netcat i listeningmodus på en angitt port, som kjører all inkommen data som kommandoer i shell? Problemstillinger her vil blant annet være om en applikasjon som ikke har root-tilgang, selv kan åpne en stengt port.

Se enda mer på sikkerhetsmekanismer. Spesielt de som er implementert i Android. Hvordan fungerer for eksempel sandbox-teknologien? Hva gjør denne sikker? File signing bør også tas en titt på.

Finne ut informasjon om kildekoden som er brukt. Hvilken linux-kjerne som er brukt og hvilke rutiner for sikkerhetsoppdateringer som eksisterer. Finst det applikasjoner på telefonen som er likt som det som kjøres på vanlige datamaskiner? Vil i disse tilfeller tilhørende sikkerhetshull også gjelde for Android?

Få frem funn som ikke bare omhandler telefoner med Android, men som gjelder alle telefoner. Dette vil ha større nytteverdi ettersom en når en større brukermasse.

http://www.mobistealth.com/ tilbyr spionering av smarttelefoner. Dette er store proof-of-concept som kunne blitt brukt på presentasjon for å vise hva som faktisk er mulig.