TRESNA LIBREEN BIDEZKO URRUNEKO ADMINISTRAZIO SEGURURAKO TEKNIKEN AZTERKETA, GNU/LINUX ETA ANDROID ERABILIZ

Asier Iturralde Sarasola

Tutorea: **Roberto Cortiñas**

Software librea graduondokoaren kurtso bukaerako ikerketa lana

2011ko ekaina-iraila

eman ta zabal zazu

universidad del país vasco

euskal herriko unibertsitatea

Edukien aurkibidea

1 Sarrera, motibazioa eta testuingurua

Interneten erabilera orokortzeak eta konexio abiadurak handitzeak urruneko administrazioaren garrantzia handitzea ekarri du. Gero eta ohikoagoa da etxeko ordenagailutik edo mugikorretik bulego edo etxeko makinara sarbidea nahi izatea, adibidez, gauden lekutik beharrezko faktura edo txosten baten kopia eskuratu ahal izateko makinaren kokapen fisikora joan beharrik izan gabe.

Gure makinetara urruneko sarbidea izatea eta administratu ahal izatea erakargarria eta ahaltsua den arren, baditu bere arriskuak ere. Guk sarbidea lor badezakegu, erasotzaile batek ere posible du berdina egitea. Beraz, guztiz garrantzitsua da teknika hauek seguruak izatea eta baimendutako erabiltzaileek bakarrik izatea sarbidea. Beste arrisku bat norbaitek sarean zehar bidaltzen ditugun datuak lortu eta irakurri ahal izatea da. Azken hau saihesteko, zifratutako konexio seguruak erabiltzea komeni da.

Aspalditik ordenagailu erabiltzailea naizen arren, ordenagailu bakarra eduki dut beti etxean eta ez dut esperientzia handirik sareekin, bideratzaileekin eta urruneko administrazioarekin. Beraz, aukera polita izan daiteke gauza berriak ikasi eta gero eta garrantzitsuagoa izango den arlo batean esperientzia pixka bat hartzeko.

Software libreak ere goranzko joera darama. 1983an GNU proiektua abiatu zenetik eta 1991an Linux nukleoaren lehen bertsioa argitaratu zenetik asko aldatu dira gauzak. Garai horietan lau katuren kontua zena denborarekin milioika lagunenganaino iritsi da, gaur egun nork ez ditu ezagutzen Firefox nabigatzailea edo Android sistema eragilea? Gero eta ohikoagoa da lizentzia libreko aplikazioen erabilera eta GNU/Linux sistema eragilea ere gero eta ezagunagoa da, batez ere erabiltzaile hasiberriei zuzendutako Ubuntu edo Linux Mint bezalako banaketei esker. Hala ere, ezin da ahaztu banaketa hauen oinarrian dagoen Debian banaketaren inguruko komunitateak urteetan zehar egindako lan eskerga, hori gabe ez baikinateke gaur egungo egoeran egongo. Munduan zehar sektore pribatu eta publikoko hainbat erakunde ere ohartzen hasiak dira software libreak eskaintzen dituen abantailez.

Gailu mugikorren artean Android sistema eragile librea hartzen ari den garrantzia azpimarragarria da. Canalys-ek kaleratu berri duen ikerketaren arabera 2011ko bigarren hiruhilabetekoan Android sistema eragiledun 51,9 milioi mugikor berri saldu dira, saldu diren smartphone guztien %48. Goranzko joera nabarmena da gainera, 2010eko bigarren hiruhilabeteko saldutakoak halako bost baita hori. Canalys-ek aztertutako 56 herrialdeetatik 35etan nagusi da Android sistema eragilea.

Android gailuak gero eta ugariagoak izateaz gain gero eta ahaltsuagoak ere badira. Urte gutxian gailu mugikorren merkatuak jauzi izugarria egin du, telefono mugikor soilak izatetik erabilera anitzeko gailuak izatera igaro baitira. Duela urte gutxiko mahaigaineko ordenagailuen mailara iristen ari dira eta etorkizunean gero eta garrantzi handiagoa izango dutela dirudi.

Ondorioz, Android eta GNU/Linux erabiliz urruneko administraziorako teknikak lantzea ideia interesgarria delakoan nago. Hau izango da lan honen helburu nagusia, hurrengo atalean zehaztuko denez.

2 Helburuak

Helburua Port Knocking, Single Packet Authorization, Wake on LAN eta urruneko administrazioko teknikak (Secure Shell -SSH-, SCP, Virtual Network Computing -VNC-, NX, ...) lantzea da, GNU/Linux-en Ubuntu banaketa duten ordenagailuak eta Android telefono mugikorrak erabiliz.

GNU/Linux-en Ubuntu banaketa erabiliz konputagailu batetik bestera sarbide segurua lortu eta administratzeko teknikak landu nahi nituzke, baita Android gailu batetik konputagailu batetara modu seguruan konektatu eta administratzeko teknikak ere. Alderantzizkoa ikertzea ere interesgarria litzateke, hau da, ordenagailu batetik Android gailura konektatu eta administratzea.

Aipatutako teknika bakoitzaren oinarri teorikoak aztertu ondoren, inplementaziorik ohikoenak nola gauzatu eta bakoitzak dituen segurtasun mugak eta arriskuak ikertu nahi nituzke.

Android sistema eragilerako aplikazioen itzulpenak nola egin ere ikasi nahi nuke eta erabiliko ditudan aplikazioak itzuli.

Android sistema eragilearen oinarriak eta Androiderako aplikazioen garapenerako erabiltzen diren tresna eta programazioaren oinarriak ere ikertu nahi ditut. Ez dut uste denborarik emango didanik baina ahal izanez gero Androiderako aplikazio batean integratu nahi nituzke aipatutako tekniketarako dauden aplikazio libreak.

3 Diseinua

Lehen pausoa ikertu nahi ditudan teknikei buruzko informazioa eta bibliografia bilatzea izango da, teknika bakoitzaren oinarri teorikoak lantzeko.

Ondoren, arlo praktikora igaroko naiz, teknika hauek lantzeko GNU/Linux-erako (konkretuki Ubuntu-rako) eta Android-erako dauden aplikazio libreak aurkitzea izango da hurrengo lana. Aplikazio bakoitzaren azterketatxo bat egindakoan probak egingo ditut:

Lehenik, VirtualBox-eko Ubuntu 10.04 makina birtualekin eta Android emulatzaileekin.

Hurrengo pausoan, D-Link DIR320 bideratzailea erabiliz sare lokalean konektatutako benetako makinak erabiliko ditut eta Android 1.6 gailu mugikorra. Horretarako lankide ohi baten laguntza izango dut, nik ordenagailu bakarra baitut eta nire mugikorra ez baita Android sistema eragileduna.

Androiderako aplikazioen itzulpen prozesua ere aztertu nahi dut eta erabiliko ditudan aplikazioak itzuli.

Azken pausoan, teknika berdinak probatuko ditut baina kasu honetan interneten zehar. NAT atzean dagoen zerbitzari batetara konektatzeko bideratzailean egin beharreko egokitzapenak landu nahi ditut.

Honekin batera, Android gailuetarako aplikazioen garapenaren oinarriak ikasten joango naiz, horretarako bi liburu erabiliko ditudalarik: “Learning Android” eta “Professional Android 2 Application Development”. Ziur aski denborarik emango ez didan arren azken helburua Android gailuetarako dauden urruneko administraziorako aplikazioak bildu eta koordinatuko dituen aplikazio bat egitea da.

4 Erabilitako teknologien deskribapena

4.1 Port Knocking

Zenbait zerbitzuk eskuragarri egon behar dute edonorentzat edonondik, adibidez 80 atakako http zerbitzuak web zerbitzari batean.

Beste zerbitzu batzuk berriz erabiltzaile jakin batzuentzat bakarrik egon behar dute eskuragarri, adibidez, 22 atakako ssh zerbitzuak. Honelako zerbitzuek, orokorrean, autentifikazio sistema bat izan ohi dute, pasahitza adibidez. Baina badira inolako autentifikaziorik erabiltzen ez duten zerbitzuak ere.

Erabilera mugatzeko ohiko metodoa suhesi bat erabiltzea da. Suhesi batek paketeak onartu edo atzera botatzen ditu paketearen ezaugarrietan oinarrituta, adibidez, IP jakin batzuetatik jasotako paketeak bakarrik onar ditzakegu. Zoritxarrez, hau ez da segurua, erasotzaile batek bere IPa mozorrotu baitezake eta gainera, kasu askotan, baimendutako erabiltzaile batek IP aldakorra izan baitezake. Suhesiak hainbat erasoren aurkako segurtasun neurri garrantzitsu eta beharrezkoak dira baina ez dira ahalguztidunak.

Segurtasun maila handitzeko modu bat suhesian autentifikazio sistema bat erabiltzea da. Baimendutako erabiltzaileak identifikatu ondoren suhesiaren arauak egoki daitezke konexioak onartzeko.

Suhesian autentifikazio zerbitzu bat erabiltzeak dituen abantailak:

- Ataka eskanerrek ezingo dituzte detektatu gure zerbitzuak. Zerbitzariaren suhesiak jasotako paketeentzat DROP politika lehenesten duenez, erasotzaileek ezingo dute jakin gure zerbitzaria martxan dagoen edo ez eta are gutxiago ze zerbitzu dauzkagun martxan.
- Segurtasun geruza gehigarri bat erasotzaileen aurrean.
    - Segurtasun arazo ezagunak dituzten zerbitzuak babesten ditu, erabiltzaile ez-baimenduentzat ikusezin eginez. Adibidez, zerbitzu jakin baten segurtasun ahulezia bat baliatzen duen exploit bat agertzen bada suhesian autentifikazio zerbitzu bat erabil dezakegu babes moduan. Autentifikazioa gainditzen duten erabiltzaileek soilik erabili ahal izango dute zerbitzua, gainerakoentzat ataka itxita egongo delarik.
    - Segurtasun kaskarra duten zerbitzu zahar edo jabedunak babestea. Administratzaileek hainbat kasutan honelako zerbitzuak erabili beste erremediorik ez dute. Pribatutasuna, osotasuna eta autentifikazioa garrantzitsuak badira VPN sistema bat erabil daiteke. Baina pribatutasuna eta osotasuna ez badira garrantzitsuak, autentifikazio sistema sinple bat nahikoa izan daiteke.
- Erabiltzailea autentifikatzea.

Suhesian autentifikazio zerbitzuak eskaintzeko modurik ezagunena “port knocking” da.

Port knocking-a ataka itxietan zehar informazioa bidaltzea ahalbideratzen duen host-to-host komunikazio teknika da. Iristen diren pakete guztiak baztertzen dituen suhesi batean zehar informazioa garraiatzean datza, bidaltzaileari erantzunik bidali gabe. Suhesiaren politika lehenetsiak DROP izan behar du eta ez REJECT/DENY, azken kasu honetan ICMP_PORT_UNREACHABLE motako pakete bat bidaltzen baitzaio bueltan bezeroari.

Hainbat aldaera daude, baina nagusiki honela sailka daitezke:

- **Ataka sekuentzia bidezko Port Knocking-a**: informazioa ataka sekuentzia bat erabiliz kodetzen da.
  - **Sekuentzia finkoa**. Lehenengo inplementazioak mota honetakoak izan ziren
  - **Sekuentzia dinamikoa**. Adb: DKS
- **Single Packet Authorization**: pakete enkriptatu bakar baten bidez kodetzen da informazioa.
- **Webknocking**: web-zerbitzari bati egindako URLak zerbitzatzeko eskaerak erabiltzen dira informazioa kodetzeko. Kasu honetan daemon atezaina web-zerbitzaria da, adibidez, apache.
- **ICMP Knock**: ICMP atezain daemon-a ICMP echo eskaeren zain egoten da eta paketeen kargaren luzerak eta iristeko ordenak ereduarekin bat datozen egiaztatzen du.

Lehen bi kasuetan datuak ataka itxietara transmititzen dira eta daemon batek aztertzen ditu, bidaltzaileari erantzunik bidali gabe. Bidalitako datuek ezarritako baldintzak betetzen dituztenean aurrez ezarritako ekintza burutzen da, adibidez, suhesiaren arauak aldatzea 22 ataka irekitzeko port knocking-a gauzatu duen bezeroari.

Helburu nagusia babes geruza gehigarri bat eskaintzea da. Erasotzaile batek gure makinaren atakak eskaneatzen baditu itxita daudela ikusiko du eta kasu askotan, etsi eta beste makina bat erasotzen saiatuko da. Modu honetan, ezagutza gutxiko erasotzaile eta eskaneatze automatikoetatik babesten gara.

Desabantailak:

- Edozein arrazoi dela eta, entzuten ari den daemona erortzen bada, konektatu ezinik geratuko gara. Atea behar bezala jo arren hildako atezainak ez digu aterik irekiko. Port Knocking inplementazio modernoek daemon monitorizatzaile bat izan ohi dute, daemon atezaina erortzen bada berriz martxan jartzen duena.
- *Network Address Translation*-ekin arazoak. NAT erabiltzen denean sare lokaletik kanpora doazen paketeei goiburua aldatzen zaie guztiek IP helbide bera erabiltzeko. Port Knocking bezeroak NAT erabiltzen duen sare lokal batetik egiten badu konexio eskaera sare lokaleko makina guztiek lortuko dute baimena, guztiek IP publiko bera baitute.
- Paketeak ordena desegokian iristea. Bezeroak paketeak ordena egokian bidaltzeak ez du ziurtatzen ordena egokian iritsiko direnik zerbitzarira. Single Packet Authorization motako Port Knocking-ean ez dago horrelako arazorik, pakete bakarra bidaltzen baita.

4.1.1 Ataka sekuentzia bidezko port knocking-a

Port knocking daemon atezainak suhesiak baztertutako paketeak aztertzen ditu eta aurrezarritako ataka zenbakietara beharrezko ordenean bidalitako pakete sekuentzia baten zain gelditzen da. Hau da, bezeroak SYN paketeak bidali behar ditu aurrezarritako ataka sekuentzia batera, adibidez, 100, 200, 300. Zerbitzariko suhesiak pakete horiek baztertu egingo ditu bidaltzaileari mezurik itzuli gabe baina, port knocking daemonak iritsitako paketeak aztertu eta beharrezko sekuentzia aurkitzen duenean aurrezarritako ekintza burutuko du, adibidez, 22 ataka irekitzea (SSH) atea behar bezala jo duen IPari. Orduan, bezeroak ssh bidez konexioa ezarri eta beharrezko lanak burutu ditzake.

Ahultasunak:

- Irekitako ataken bila dabiltzan ezagutza gutxiko erasotzaileetatik ('script kiddie' deituak) eta atakak eskaneatzen dituzten bot-etatik babesteko nahikoa izan daiteke inplementazio soil hau, modu honetan erasoen portzentaia handi bat atzera botako dugularik. Baina erasotzaile prestatuagoek erraz gaindi dezakete segurtasun neurri hau. Bezero eta zerbitzariaren arteko trafikoa monitoreatzeko gauza den erasotzaile batek erraz lor dezake erabilitako portu sekuentzia eta errepikapenezko eraso bat burutu. Hau eragozteko portu sekuentzia dinamiko bat erabil daiteke.
- Ordena desegokian iristen badira paketeak port knocking daemonak ez du sekuentzia ezagutuko. Nahiko erraz gerta daiteke hau, zerbitzariak ez baitu erantzunik itzultzen eta bidalitako paketeek sarean zehar bide desberdinak jarrai baiditzakete, zerbitzarira iristeko denbora desberdina erabiliz.
- Erasotzaile batek erraz sabotea dezake bezeroak bidalitako portu sekuentzia, nahikoa du bezeroa pakete sekuentzia bidaltzen ari denean, IP berdina erabiliz, pakete gehigarri bat bidaltzea. Itxuraz pakete guztiak (bezeroarenak eta erasotzailearenak) IP berdinetik datozenez zerbitzariak portu sekuentzia desegoki bat detektatuko du eta ez du erantzungo.
- Zerbitzuaren ukatze (*Denial Of Service*) motako erasoak

4.1.1.1 Inplementazio librea: Knock eta Knockd

Knockd port knocking zerbitzari bat da. Ethernet (edo PPP) interfaze batean zehar igarotzen den trafiko guztia entzuten du, bezeroak ataka jakin batzuetara bidalitako TCP (edo UDP) pakete bidezko “knock” edo ate jotze sekuentzia baten zain. Portu hauek itxita egoten dira, baina knockd zerbitzariak lotura geruza mailan entzuten duenez gai da portu itxietara bidalitako trafikoa antzemateko. Beharrezko ataka sekuentziara bidalitako pakete sekuentzia jasotzean bere konfigurazio fitxategian definitutako komandoa egikaritzen du, orokorrean bezeroari ataka bat irekitzeko.

Knock port knocking bezeroa da eta knockd-ri ataka sekuentzia jakin batetara paketeak bidaltzeko prestatua dago. Hala ere, TCP paketeen sekuentzia sinpleak bidaltzeko ssh, telnet edo netcat ere erabil daitezke.

Software librea dira.

Webgunea: http://www.zeroflux.org/projects/knock

Iturburu kodea: http://knockd.sourcearchive.com/

4.1.2 Single Packet Authorization

Izenak dioen bezala Single Packet Authorization-ean pakete bakarrean kodetzen da informazioa. Pakete honi Autorizazio Paketea deitzen zaio eta UDP edo ICMP motakoa izan ohi da.

Port Knocking tradizionalak bezala bezero/zerbitzari egitura dauka SPAk eta zerbitzariak pasiboki monitorizatzen ditu suhesiak atzera botatzen dituen paketeak. Baina ohiko port knocking-ean ez bezala, SPAn datu transmisioa aplikazio geruzan ematen da. Modu honetan pakete bakoitzean 2 byte bidali ordez 1500 byte bidal ditzakegu gutxienez Ethernet sare bat erabiltzen ari bagara.

Autentifikatzeko pakete bakarra erabiltzeak baditu abantailak portu sekuentzia bidez kodetutako port knocking-aren aldean, adibidez, SPA erabiltzean ez dago ordenaz kanpo iritsitako paketeen arazorik. Lehen inplementazioa fwknop izeneko programa izan zen 2005eko maiatzean.

Inplementaziorik sinpleenean, data-zigilu (*timestamp*) bat (errepikapen erasoak saihesteko), bezeroaren IPa eta pasahitz bat kodetu eta pakete batean bidal daitezke zerbitzarira. Adibidez,

- Data-zigilua: 201105032224
- Bezeroaren IPa: 74.125.230.210
- Pasahitza: pasahitzsekretua

Adibide sinple honetan, zerbitzaria erabiltzaile bakarrarentzat dago konfiguratuta eta komando bakarra egikarituko du, adibidez 22 ataka 5 minutuz irekitzea bezeroaren IParentzat. Zerbitzariak balioztatutako paketeen erregistro bat gordeko du, errepikapenezko erasoak saihesteko. Informazioaren kodetzea MD5 hash bidez emango da, adibidearekin jarraituz:

MD5("201105032224:74.125.230.210:pasahitzsekretua")

Hash hau UDP pakete batean txertatu eta zerbitzarira bidaliko dugu ondoren. Zerbitzariko SPA daemon atezainak hash-a birkalkulatuko du, pasahitza (ezagutzen du), momentuko data-zigilua (YYYYMMDDhhmm) eta eskaera egin duen bezeroaren IPa erabiliz. Kalkulatutako hash-a eta jasotako paketearenak berdinak badira 22 ataka irekiko dio bezeroaren IPari. Paketearen hash-a aurretik jasotako hash-en zerrendan badago edo jasotako hash-a eta kalkulatutakoa bat ez badatoz ez du ezer egingo.

4.1.2.1 Inplementazio librea: Fwknop

Fwknop 2004ean sortu zuten eta jatorrizko inplementazioak sistema eragile atzemate pasiboa (*passive OS fingerprinting*) eta port knockinga uztartzen zituen, modu honetan sistema eragile mota jakin batetik, adibidez linux 2.6etatik, datozen port knock-ak bakarrik onartzea ahalbideratzen zuen.

2005eko maiatzean Single Packet Authorization-en lehen inplementazioa gehitu zioten eta denborarekin programaren aukera lehenetsia bihurtu da.

fwknop-ek aplikazio geruzan zehar bidalitako paketeek honelako egitura dute:

- **Ausazko 16 byte**: Honi esker port knocking tradizionalaren arazo arrunt bat konpontzen da, deien errepikapena. Bezeroak bidalitako pakete bakoitzak ausazko 16 byte dituenez eta zerbitzariak onartutako paketeen MD5 baturak gordetzen dituenez, guk egindako deia erasotzaile batek errepikatuko balu zerbitzariak atzera botako luke eta gertaeraren berri idatzi syslog-ean, berak bidalitako paketearen MD5 batura guk bidalitakoaren berdina izango bailitzateke.
- **Bezeroaren erabiltzaile izena**: Erabiltzaile desberdinei baimen desberdinak eman ahal izateko gehitzen da.
- **Bezeroaren data-zigilua (timestamp)**
- **fwknop-en bertsioa**: Bertsio zaharrak erabiliz bidalitako paketeen bateragarritasuna ziurtatzeko erabiltzen da.
- **Modua (sarrera edo komandoa)**: Bezeroak zerbitzu batera sarrera nahi duen edo komando bat egikaritu nahi duen zehazten du.
- **Sarrera (edo komando katea)**: Adibidez, TCP 22 atakara sarrera nahi bagenu <IP>,tcp/22 zehaztu beharko genuke, non, <IP> bezeroaren IP zenbakia den.
- **MD5 batura**: Enkriptatu gabeko paketearen MD5 batura. Zerbitzariak jasotako paketeak aldaketarik ez duela jasan egiaztatzeko balio du.

SPA pakete bateko hainbat eremuk luzera aldakorra dute eta : ikurra erabiltzen da mugatzaile bezala (eremuak base64 erabiliz kodetuta daudenez ez dago arazorik eremu barruko : ikurrekin). fwknop bezeroak aipatutako paketea osatu ondoren, osorik enkriptatzen du 128 bit-eko gako partekatua duen Rijndael zifraketa simetriko bidez edo GnuPG bidez sortutako 2048 bit-erainoko gako publiko/pribatu parea duen ElGamal algoritmo asimetrikoarekin. Ondoren 62201 UDP atakan zehar bidaltzen du. Erabilitako portu zenbakia erraz alda daiteke komando lerroan --Server-port argumentua erabiliz.

Webgunea: http://www.cipherdyne.org/fwknop/

Kode biltegia: http://www.cipherdyne.org/fwknop/download/

4.1.3 Web Knocking

Oinarrizko ideia ataken ordez web orriak erabiltzea da. Hori dela eta, beharrezkoa da helburuko makinan web zerbitzari bat edukitzea, adibidez, apache. Zerbitzari honi web orri eskaera sekuentzia jakin bat egingo diogu. Web zerbitzariak eskaera hauek log batean gordeko ditu, hori da konfigurazio lehenetsia apachen. Sekuentziako azken web orria finkoa da eta web zerbitzariaren logaren azterketa hasierazten du. Beharrezko web orri sekuentzia aurkitzen badu script bat egikaritzen da, adibidez, 22 ataka irekitzeko bezeroari.

Erasotzaile batek web sekuentzia usna dezake, port knocking arruntean portu sekuentziarekin gertatzen den bezala, beraz beharrezkoa da web sekuentzia dinamiko bat erabiltzea. Hau bi modutan gauza daiteke:

1. Sekuentzia guztiz dinamikoa. Erabiltzen den aldi bakoitzean web orri sekuentzia desberdina erabiltzen da kasu honetan, logaren azterketa abiarazten duen orria ere aldakorra delarik.
2. Sekuentzia erdi dinamiko erdi estatikoa.

Segurtasun ikuspuntutik sekuentzia guztiz dinamikoa erabiltzea egokiagoa da, usnatzaileek ez baitute eredu errepikakorrik aurkituko. Baina honek bezeroari gauzak zailtzen dizkio, aldi bakoitzean erabili beharko duen sekuentzia aldakorra zein den jakin beharko baitu. Errazagoa da partzialki estatikoa den sekuentzia bat gogoratzea, baina honek erasotzaileari ere gauzak errazten dizkio. Zati estatikoa bat datorrela ikusi ondoren web zerbitzariak zati dinamikoa galdera-erantzun joku baten bidez gauza dezake. Galdera erantzunak algoritmo baten bidez lotuak egon daitezke (adb, “Zenbat da 6 bider 7?”) edo erabiltzailearen kontestuari lotuak (adb, “Non jaio zinen?”).

Web knocking inplementazioak:

- **Webknocking v1.0.2**

  http://lebelt.info/old/content/_files/webknocking_v1.0.2.tar.gz

- **wwwknock 1.2**

  http://forums.freebsd.org/attachment.php?s=0069f777f512b4147529b8b700760c5e&attachmentid=416&d=1253842576

4.1.4 ICMP Knocking

Aldaera honetan TCP edo UDP paketeen ordez ICMP ECHO_REQUEST datagramak dira jasotzen direnak eta bere karga eta iristeko ordena dira autentifikaziorako aldagaiak.

Abantaila nagusia bezeroak ez duela programa berezirik erabili behar da, PING komandoarekin nahikoa da.

ICMP Knocking inplementazioa:

- **ICMP Knock Server**: Python lengoaian idatzitako zerbitzaria. BSD lizentzia.
  http://homework.nwsnet.de/products/0d09_icmp-knock-server

4.1.5 Segurtasuna

4.1.5.1 Port Knocking-a iluntasun/ezkutatze bidezko segurtasuna al da?

Port Knocking-ari egin izan zaion kritiketako bat iluntasun/ezkutatze bidezko segurtasuna motako teknika bat izatea da.

Ikus dezagun adibide baten bidez iluntasun bidezko segurtasuna zer den. Demagun sistema-administratzaile batek informazio garrantzitsua duen orri bat ezkutatzeko URL ilun/zail bat erabiltzen duela, URLa ezagutzen ez duen inork aurkituko ez duelakoan. Adibidez,

http://www.adibidea.com/0832948242349isfdjou784932ou4o32u4o23j4l23j.html

Hau iluntasun bidezko segurtasuna da eta ez da segurua, URL sekretu hau bertara konektatzen diren bezeroen ordenagailuetako log fitxategietan, nabigatzailearen historian, eta abarrean agertuko da eta ordenagailu bera erabiltzen duen baina orri hau ikusteko baimenik ez duen norbaitek helbidea aurkitu eta orria ikus dezake. Ezkutuan entzuten ari den erasotzaile batek ere lor dezake URLa, http erabiltzean testu lau bezala bidaltzen baitira eskaerak. Googleren bilatzailean agertzeko arriskua ere badago. Azken finean, metodo hau ez da SSL gaituta duen orri batean karaktere kopuru bera duen pasahitz bat erabiltzearen baliokidea.

Hau egiteak txorakeria dirudi, nork egingo du ba horrelakorik? Baina Erresuma Batuko O2 operadoreak MMS bidez argazkiak jasotzeko gauza ez ziren mugikor moten erabiltzaileei, adibidez Apple-ren iPhone 3G-ren erabiltzaileei, argazkiak helarazteko teorian jasotzaileak bakarrik ezagutu beharko lukeen URL ilun batean publikatzen zituen argazkiak eta helbidea e-posta bidez bidali. Adibide hau bezalakoak ziren URL hauek:

http://mediamessaging.o2.co.uk/mms2legacy/showMessage2.do?
encMmsId=F1ABCF6D326A3F65

Baina argazki horiek eskuragarri zeuden edonorentzat, nahikoa zen googleren bilatzailean inurl:mms2legacy site:o2.co.uk jartzea. Argazki hori ikusi ondoren beste zerbitzari batetara nabigatzean referrer tag-ean agertuko da URL “sekretua” eta zerbitzariaren referrer log-ean gordeko da eta hemendik Googlebot-ak irakurriko du eta bilaketetan agertuko da. Oso zaila da web zerbitzari bat sekretuan mantentzea, egin daitekeena pasahitzez babestea da.

Googleren bilatzailea erabili gabe ere aurkitu zitezkeen argazkiak, URL-etan 16 digituko zenbaki hexadezimaleko IDak (64 bit) erabiltzen zituen 02-k eta konbinazio guztiak probatzeko scriptak erabiltzen zituen jendeak argazkiak ikusteko.

Ikusi dugun bezala iluntasun bidezko segurtasuna sekretu batean oinarritzen da, adibideko URLan bezala eta ez dauka sarbide kontrolik edo defentsa mekanismo aktiborik. Sekretu bakar batean oinarritzen denez ezinezkoa da kontrol zehatzik ezartzea.

Port knocking-a ere sekretu batean oinarritzen da, ate jotze sekretuan. Baina sarbide kontrola ezartzeko aukera eskaintzen du. Port knocking-eko daemonak sarbide saiakeren zerrenda gordetzen du log batean. Indar gorrizko eraso ahaleginak modu erraz batean detektatu eta saihets daitezke.

Port Knocking-a segurtasun geruza gehigarri bat da eta ez ohizko segurtasun neurrien ordezkoa.

4.1.6 Bibliografia

Ataka sekuentzia bidezko Port Knocking-a

- http://en.wikipedia.org/wiki/Port_knocking
- **"An Analysis of Port Knocking and Single Packet Authorization"**, Sebastien Jeanquier, MSc Thesis (2006) (http://www.securitygeneration.com/wp-content/uploads/2010/05/An-Analysis-of-Port-Knocking-and-Single-Packet-Authorization-Sebastien-Jeanquier.pdf)
- **Analysis of Port Knocking Mechanism**, Mehmet Ülkem Demercioglu, Master Thesis (2009) (http://acikarsiv.atilim.edu.tr/browse/259/272.pdf)
- http://www.portknocking.org/
- http://www.aldabaknocking.com/documentation

Single Packet Authorization

- http://www.cipherdyne.org/fwknop/
- http://www.cipherdyne.org/fwknop/docs/SPA.html
- "Single Packet Authorization", Michael Rash, The Linux Journal, April 2007 (http://www.linuxjournal.com/article/9565)
- "An Analysis of Port Knocking and Single Packet Authorization", Sebastien Jeanquier, MSc Thesis (2006) (http://www.securitygeneration.com/wp-content/uploads/2010/05/An-Analysis-of-Port-Knocking-and-Single-Packet-Authorization-Sebastien-Jeanquier.pdf)
- https://help.ubuntu.com/community/SinglePacketAuthorization

Web Knocking

- http://lebelt.info/old/content/_files/webknocking_v1.0.2.tar.gz
- http://forums.freebsd.org/attachment.php?s=0069f777f512b4147529b8b700760c5e&attachmentid=416&d=1253842576

ICMP Knocking

- http://homework.nwsnet.de/products/0d09_icmp-knock-server

Segurtasuna

- http://www.portknocking.org/view/about/obscurity
- http://www.portknocking.org/view/faq/security
- http://www.portknocking.org/view/about/critique
- http://stackoverflow.com/questions/4833314/are-secret-urls-truly-secure
- http://stackoverflow.com/questions/1486673/i-want-to-use-security-through-obscurity-for-the-admin-interface-of-a-simple-webs
- http://docbug.com/blog/archives/000780.html
- http://www.informationweek.com/news/mobility/security/209101313
- http://www.mattcutts.com/blog/toolbar-indexing-debunk-post/

4.2 Wake-On-LAN

Wake-on-LAN (WOL) urrunetik ordenagailu bat piztea ahalbideratzen duen estandarra da. Wi-fi bidez gauzatzen bada Wake on Wireless LAN (WOWLAN) deitzen da.

Lehen garapena AMD/HP-k egin zuten eta sare-pakete berezi bat erabiliz gauzatzen da. Gaur egungo ordenagailu gehienek onartzen dute WoL.

Sareko interfaze txartelak (*Network Interface Card*, NIC) paketeak aztertzen ditu eta baliozko WOL pakete bat detektatzean ordenagailua pizten du.

Wake-on-LAN erabiltzeko beharrezko baldintzak:

- WOL onartzen duten txartel nagusi (*motherboard*) eta BIOSa.
- Sareko txartel bateragarri bat.
- Sistema eragilearen konfigurazio egokia. Sareko Interfaze Txartelak (NIC) martxan jarraitu behar du ordenagailua itzalita dagoenean.

Oinarrizko WoL pakete batek ondoko osagaiak ditu:

- Goiburu sinple bat
- Piztu nahi dugun makinaren MAC (*Media Access Control*) helbidea 16 aldiz errepikatuta

Edozein sare edo garraio mailako protokolo erabil daiteke pakete hau bidaltzeko. Kasurik ohikoena UDP pakete bat erabiltzea da.

WOL paketeen mugak:

- Ordenagailu bakarrarentzat balio du pakete bakoitzak. Sareko Interfaze Txartelak erraz prozesa ditzan WOL paketeak oso sinpleak direlako gertatzen da hau.
- Hardwarearen MAC helbidea behar da. WOL-ek protokolo geruza baina maila baxuagoan lan egiten duelako gertatzen da hau. Maila honetan IP helbideek eta DNS izenek ez dute garrantzirik.
- Ezin dugu ziurtatu paketea helburura iritsi den. UDP paketeak erabiltzeagatik gertatzen da hau.
- Arazoak sare lokaletik kanpo. WOL difusio paketeak erabiliz gauzatu ohi delako gertatzen da hau. Pakete hauek ez dira bideratzen eta ondorioz sare lokalaren muga ez dute gainditzen.

Wake-on-LAN teknologiaren inplementazioa gogaikarria izan daiteke, BIOSa, sare txartela, sistema eragilea eta kasu batzuetan bideratzailea egokitu beharko baititugu behar bezala lan egin dezaten. Kasu batzuetan makina guztiz itzalita badago piztuko da baina esekita edo hibernatzen badago berriz ez. Ez dago argi sareko txartel interfaze (NIC) batzuek espero dituzten pakete magikoak zein motatakoak diren ere.

Honelako arazoak daudenean, pakete analizatzaileen erabilera oso egokia da, ordenagailua piztuta dagoen bitartean bere NICak pakete magiko bat ikusi duela egiaztatzea errazten baitute. Pakete magiko bera erabil daiteke ordenagailua offline egoeratik pizteko, modu honetan sare mailako arazoak hardware mailako arazoetatik isola ditzakegu. Kasu batzuetan paketea PC jakin bati zuzendua izan den edo difusio helbide batera bidali den egiazta daiteke eta paketearen barneko egitura ikus dezakegu gainera.

4.2.1 Pakete magikoa

Difusio-marko (*Broadcast frame*) bat da, bere kargan (*payload*) 6 aldiz 255 daukana (FF FF FF FF FF FF hexadezimalean) eta jarraian helburuko konputagailuaren 48-biteko MAC helbidea 16 aldiz errepikatuta. Adibidez, helburuko konputagailuaren MAC helbidea 01:02:03:04:05:06 balitz ondoko katea lortuko genuke:

FFFFFFFFFFFF010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506010203040506

Pakete magikoa eskaneatzean aipatutako katea da bilatzen den bakarra eta ez da pakete osoa analizatzen edo parseatzen, ondorioz edozein sare edo garraio protokolo erabiliz bidal daiteke. Hala ere, orokorrean 7 edo 9 atakara bidalitako UDP datagrama izan ohi da.

Pakete magiko estandar batek ondoko mugak ditu:

- Helburuko konputagailuaren MAC helbidea behar du (eta behar bada SecureOn pasahitza)
- Ez du helburura iritsi denaren egiaztapenik bidaltzen jatorrira.
- Erabilpena sare lokalera mugatuta egon daiteke.
- Helburuko konputagailuaren hardwareak Wake-on-Lan onartu behar du.

Wake-on-Lan oso soila izateko diseinatua dago, helburuko sare interfaze txartelak (NIC) azkar eta argindar kontsumo minimoarekin prozesa dezan.

Wake-on-LAN-ak protokolo geruza baina maila baxuagoan lan egiten duenez MAC helbidea beharrezkoa da eta IP helbideak eta DNS izenek zentzua galtzen dute.

4.2.2 Azpi-sareetara bideratutako difusioak

Orokorrean difusio paketeak ez bideratzea da Wake-on-LAN estandarraren mugapen garrantzitsuenetako bat, hau dela eta, teknika hau sare handietan edo interneten zehar erabiltzea zailtzen da. Azpisareei zuzendutako difusioa (*Subnet Directed Broadcast*, SDB) erabil daiteke muga hau gainditzeko, horretarako beharrezko izan daiteke bideratzailearen konfigurazioan aldaketak egitea. Azpisareei zuzendutako difusio paketeak sareko pakete arrunt bezala tratatzen dira azken bideratzaile lokalera iritsi arte. Orduan bideratzaile honek benetako difusio pakete bihurtzen ditu. Teknika honen bidez difusio bat urruneko sare batean has daiteke baina tarteko bideratzaile guztiak SDBa bideratzera behartzen ditu. Honelako sare bat SDB paketeak birbidaltzeko prestatzean iragazki bat jarri behar da, guk nahi ditugun paketeak bakarrik onartzeko, bestela DDoS motako erasoen aurrean babesgabe geratuko baikara, adibidez, *Smurf Attack* motako erasoen aurrean.

4.2.3 Segurtasuna

Sare txartel interfaze batzuek “SecureOn” izeneko segurtasun ezaugarria onartzen dute. Honi esker, erabiltzaileak gai dira NICean 6 byteko pasahitz hexadezimal bat gordetzeko. Kasu hauetan, erabiltzaileak pasahitz hau gehitu behar dio pakete magikoari eta NICak pasahitza eta MAC helbidea egokiak badira bakarrik esnatuko du makina. Segurtasun neurri honi esker asko jaisten da indar gorrizko eraso arrakastatsuen arriskua, saiatu beharreko aukerak 48 bitetik (6 byte) 2^{96}raino igotzen baitira MAC helbidea guztiz ezezaguna bada.

Hala ere, NIC eta bideratzaile gutxi batzuek onartzen dituzte honelako segurtasun neurriak. Edonola ere, erasotzaile batek Wake-on-LAN-aren segurtasun neurriak gainditua ere, gehienez ere ordenagailua piztea lortuko du, ez du besterik gabe sarbiderik lortuko.

4.2.4 DD-WRT firmware-a

DD-WRT Broadcom edo Atheros txip diseinuetan oinarritutako ieee802.11a/b/g/h/n motako haririk gabeko bideratzaileetan firmware ofiziala ordezkatzeko hirugarrenek egindako Linux-en oinarritutako firmware-a da, GPL lizentziapean argitaratua. Hainbat arrazoirengatik egin ohi da hau, horien artean jatorrizko firmware-ak eskaintzen ez dituen ezaugarriak gehitzea, hala nola, Kai sareak, daemonetan oinarritutako zerbitzuak, IPv6, Harigabeko Banaketa Sistemak (*Wireless Distribution System*), RADIUS, *quality of service* aurreratua, *radio output power control*, erlojua azkartzeko ahalmena (*overclocking*) eta *Secure Digital Card* hardware aldaketentzako software euskarria.

BrainSlayer-rek mantentzen du eta www.dd-wrt.com-en ostatatua dago. Lehen bertsioak Sveasoft Inc-en Alchemy Firmware-an oinarrituta zeuden eta Alchemy Firmware berriz GPL lizentziadun Linksys firmwarean eta beste software libreko hainbat proiektutan zegoen oinarrituta. Sveasoft-ek jabedun lizentzietara pasatzeko erabakia hartu zuenean sortu zen DD-WRT.

v22 bertsiorarte Linksys-en jatorrizko firmwarean oinarritutako Sveasoft-en Alchemy firmware-an oinarrituta zegoen arren v23 bertsioan ia erabat berridatzia izan zen kodea. DD-WRT-ren azken bertsioa, v24, proiektu guztiz berria da. DD-WRT-k bideratzaileen jatorrizko firmware-ek eskaintzen ez dituzten hainbat ezaugarri aurreratu ditu.

DD-WRT doan eskura daiteke, baina ordainpeko bertsioak ere badaude.

Buffalo Technology eta beste enpresa batzuek DD-WRT-ren bertsio egokituak erabili izan dituzte beren bideratzaileetan.

Webgunea: http://www.dd-wrt.com/

4.2.4.1 Bertsioak

- DD-WRT v23 Service Pack 1 (SP1): 2006ko maiatzaren 16an argitaratua. Kodearen zati handi bat hutsetik berridatzia izan zen bertsio honetan eta ezaugarri berri asko gehitu ziren.
- DD-WRT v23 Service Pack 2 (SP2): 2006ko irailak 14an argitaratua. Interfazea berritu zen eta zenbait ezaugarri berri gehitu. Bideratzaile gehiagorentzako euskarria gehitu zen.
- DD-WRT v24: 2008ko maiatzaren 18an argitaratua. SSID eta zifratze protokolo desberdindun 16 interfaze birtual onartzen ditu gehienez. Sistema gehiagorentzako euskarria, hala nola zenbait PowerPC, IXP425-ean oinarritutako bideratzaile plakak, Atheros WiSOC eta X86an oinarritutako sistemak. Flash memoria baxuko bideratzaileentzako euskarri mugatua ere eskaintzen du, adibidez, WRT54Gv8 edo WRT54GSv7.
- DD-WRT v24 Service Pack 1 (SP1): 2008ko uztailaren 26an argitaratua. dnsmasq-entzako DNS segurtasun arazoarentzako presazko konponketa, Site Survey segurtasun konponketak, pasahitz luzeagoak eta OpenVPN konfigurazio malguak. Hardware gehiagorentzako euskarria, tartean WRT300 v1.1, WRT310N, WRT600N, Tonze AP42X Pronghorn SBC, Ubiquiti LSX eta Netgear, Belkin eta USR gailuak.
- DD-WRT v24 Service Pack 2 (SP2): Garapenean.

4.2.4.2 Lizentziari buruzko eztabaida

2006ko uztailean DD-WRT Denda jabedun lizentziadun zatiak zituen bertsio komertzial bat eskaintzen hasi zen. Funtzionalitate gehitua eskaintzen du bertsio komertzial honek, adibidez, erabiltzaile bakoitzeko banda zabalera kontrola. Zenbaitek salatu zuenez, garai berean lizentziaren testua aldatu zen osagai batzuk ez zutela GPL lizentzia zehazteko. Garatzaileak salaketa hauei

erantzun zien bere blogean eta iturburu kodea eskuragarri dagoela argitu. Lizentziaren testua berriz ere GPL izatera itzuli zen.

4.2.4.3 Segurtasun arazoak

Bulgariar hacker batek urrunetik komandoak exekutatzea ahalbideratzen zuen ahultasun bat aurkitu zuen DD-WRT-ren HTTPd zerbitzarian 2009ko uztailaren 21ean eta milwOrm-en jakinazi ondoren ordu gutxi barru konpondua izan zen. Hala ere, V24-ren SP2 aurreko 12533 eraikitzearen aurreko bertsio guztiak babesgabe daude exploit honen aurrean.

4.2.5 DynDNS

Doako DNS dinamiko zerbitzua da. URL helbide baten bidez IP dinamikoa duten interneterako konexioetara modu errazean konektatzeko modua ematen du. Bezeroaren ordenagailu edo bideratzailean eguneraketa bezero bat instalatzen da eta IP helbidea aldatzen denean zerbitzariari jakinarazten dio, IP eta URL-aren arteko lotura eguneratuta mantenduz.

Webgunea: http://dyn.com/

4.2.6 Inplementazioa librea: gWakeOnLan

gWakeOnLan Wake on LAN teknika erabiliz ordenagailuak pizteko GTK+ tresna da.

GPLv2 lizentziapean kaleratutako software librea da.

Kode biltegia: http://code.google.com/p/gwakeonlan/

4.2.7 Bibliografia

Wake on LAN

- http://en.wikipedia.org/wiki/Wake_on_lan
- http://www.depicus.com/wake-on-lan/what-is-wake-on-lan.aspx

DD-WRT

- http://en.wikipedia.org/wiki/DD-WRT
- http://www.dd-wrt.com/wiki/index.php/What_is_DD-WRT%3F
- The DD-WRT Controversy By Aaron Weiss:
  http://www.wi-fiplanet.com/columns/article.php/3816236
- DD-WRT-ren bertsio berezia:
  http://www.dd-wrt.com/shop/catalog/product_info.php?cPath=22&products_id=31
- Garatzailearen erantzuna:
  http://dd-wrt.blogspot.com/2007/02/as-we-see-jeremy-collake-is-still-alive.html

DynDNS

- http://en.wikipedia.org/wiki/DynDNS

4.3 SSH

SSH edo Secure Shell sareko protokolo bat da, bi makinen arteko datu truke segurua ahalbideratzen duena.

Bi bertsio nagusi daude:

- SSH1 edo SSH-1
- SSH2 edo SSH-2

Batez ere GNU/Linux-en eta Unix-en oinarritutako beste sistemetan erabiltzen bada ere plataforma-anitza da eta hainbat sistema eragiletarako inplementazioak daude.

Telnet eta beste urruneko shell protokolo ez-seguruak ordezkatzeko helburuarekin sortua izan zen, hauek informazioa testu lau bezala bidaltzen baitzuten, pasahitzak barne. SSH protokoloan berriz informazioa zifratuta bidaltzen da datuen konfidentzialtasuna eta osotasuna ziurtatzeko segurtasunik gabeko sare batean zehar.

Bezero-zerbitzari eredua jarraitzen du. TCP 22 da ataka lehenetsia, baina kasu askotan beste ataka bat erabiltzen da segurtasun neurri gehigarri bezala.

Kasurik sinpleenean pasahitz bidezko autentifikazioa erabiltzen da, zifratzea automatikoki sortutako gakoen bidez ematen delarik. Lehen aldiz konektatzean konexioa baimentzeko eskatuko zaigu eta hortik aurrera autentifikazioa ziurtatzeko erabilitako gakoak gogoratzen ditu. Pasahitz bidezko autentifikazioa desgai daiteke eta gako publikodun kriptografia erabili, ondoren azalduko den bezala.

Segurtasun maila altuago bat nahi izanez gero, SSH protokoloan gako publikodun kriptografia erabil daiteke urruneko konputagailu bat autentifikatzeko eta behar izanez gero erabiltzailea ere autentifikatzeko. Edozeinek sor dezake gako pare bat (publikoa eta pribatua). Gako pribatua duten erabiltzaileak dagokion gako publikoa duten ordenagailu guztietara sartu ahal izango dira, beren gako pribatua sekretuan gordez, autentifikazioa gako pribatuan oinarritu arren gako pribatua ez baita autentifikazio prozesuan zehar inoiz bidaltzen.

SSH-k egiaztatzen duena gako publikoa eskaintzen duen pertsonak dagokion gako pribatua ote duen da. Horregatik garrantzitsua da gako publikoak egiaztatzea onartu aurretik, bestela erasotzaile bat erabiltzaile arrunt bat bezala onartzen ariko baikara.

GNU/Linux-en gakoak baimendutako erabiltzailearen Home katalogoan gordetzen dira, ~/.ssh/authorized_keys fitxategian hain zuzen. Fitxategi hau aldatzeko baimena jabeak soilik izan behar du, bestela SSH-k ez du onartuko.

4.3.1 Historia eta garapena

1.x bertsioa

1995ean Tatu Ylönen izeneko Helsinkiko Unibertsitate Teknologikoko ikerlariak diseinatu zuen protokoloaren lehen bertsioa, gaur egun SSH-1 bezala ezagutzen dena. Unibertsitateko sarean gertatutako usnaketa edo *sniffing* bidezko pasahitz lapurretak saihestea izan zen horretara bultzatu zuen arrazoia. Helburua ordura arte erabiltzen ziren rlogin, TELNET eta rsh protokoloentzat ordezko seguru bat sortzea izan zen. 1995eko uztailean freeware bezala argitaratu zuen bere inplementazioa Ylönen-ek eta bere erabilera azkar zabaldu zen eta 1995eko urte bukaerarako 20000 erabiltzaile inguru zituen 50 herrialdetan banatuta.

1995eko abenduan, Ylönen-ek SSH Communications Security sortu zuen SSH garatu eta merkaturatzeko. Jatorrizko garapenak software libreko hainbat liburutegi erabiltzen zituen, adibidez

GNU libgmp baina SSH Communications Security-k kaleratutako ondorengo bertsioetan gero eta software itxiagoa bihurtzen joan zen.

2000. urterako 2 milioi erabiltzaile inguru zituen SSH-k.

SSH 1.5ak zuen segurtasun ahulgune nabarmen bat aurkitu zen 1998an. *SSH Compensation Attack Detector* izeneko zuzenketa bat gehitu zitzaien inplementazio gehienei arazo hau konpontzeko. Baina ondoren errore berri bat agertu zen eguneratutako inplementazio horietan, zenbaki osoen gainezkatze motakoa, erasotzaileei SSH daemonaren pribilegio mailan (orokorrean root) kodea exekutatzea ahalbideratzen zuena.

2001eko urtarrilean beste segurtasun ahulgune bat aurkitu zen IDEA-zifraketadun saio baten azken blokea aldatzea ahalbideratzen ziena erasotzaileei. Hilabete berean beste segurtasun ahulgune bat aurkitu zen, zerbitzari maltzur bati beste zerbitzari bati bezero autentifikazioa birbidaltzea ahalbideratzen ziona.

SSH-1-ek zituen diseinu akatsek eragindako segurtasun eskasa dela eta zaharkitutzat hartzen da eta ez erabiltzea gomendatzen da. Zerbitzari eta bezero moderno gehienek SSH-2 erabiltzen dute.

1.99 bertsioa

2.1 bertsioa kaleratu ondoren, 2006ko urtarrilean RFC 4253an zehaztu zenez 2.0 eta bertsio zaharragoak, biak, onartzen dituen SSH zerbitzariak 1.99 protobertsioa erabiliz izendatu behar dira. Ez da benetako bertsio bat baizik eta atzeranzko bateragarritasuna identifikatzen duen metodo bat.

OpenSSH eta OSSH

1999an software librearen aldeko garatzaileek lizentzia libreko azken bertsioa, hau da, 1.2.12-a, hartu eta Björn Grönvall-en OSSH sortu zuten. Pixka bat beranduago, OpenBSD-ren garatzaileek oinarri bezala OSSH erabiliz OpenSSH sortu zuten, OpenBSD-ren 2.6 argitalpenarekin kaleratu zena. Bertsio honetatik, gainerako sistema eragileekin bateragarria zen adar berri bat sortu zen. 2005erako OpenSSH zen SSHren inplementazio zabalduena, hainbat sistema eragilerekin batera banatzen zelarik. OSSH berriz atzera gelditu zen. OpenSSH-ren garapenak aurrera jarraitu du eta orain 1.x eta 2.x bertsioekin bateragarria da.

2.x bertsioa

2006an protokoloaren bertsio berrikusi bat, SSH-2, onartu zen estandar bezala. Bertsio hau bateraezina da SSH-1ekin. Diffie-Hellman gako trukeak eta mezu autentifikatze kode bidezko osotasun egiaztatze sendoak segurtasuna hobetzea ekarri zuten. SSH-2-k ekarritako berritasunen artean zegoen SSH konexio bakarra erabiliz hainbat shell saio hasi ahal izatea.

2008ko azaroan SSH-ren bertsio guztiei eragiten zien ahultasun bat aurkitu zen, garai hartan zifratze estandar lehenetsia zen CBC bidez kodeketatik 32 biterainoko testu lauak lortzea ahalbideratzen zuena.

4.3.2 Erabilerak

- Urruneko ostalari batean shell saio bat hastea (Telnet eta rlogin ordezkatuz).
- Urruneko ostalari batean komando bakar bat exekutatzea (rsh ordezkatuz).
- Fitxategi transferentzia segurua urruneko ostalariekin.
- rsync-ekin batera erabilita segurtasun kopiak egitea edo fitxategien kopiak egin eta ispiluak (*mirror*) sortzea modu eraginkor eta seguruan.

- Ataka birbidalketak edo tunelatzeak egitea (ez da nahasi behar *Virtual Private Network*-ekin (VPN)).
- Zifratutako VPN bezala erabiltzea. OpenSSH zerbitzari eta bezeroek soilik eskaintzen dute aukera hau.
- Urruneko ostalari batetik X birbidaltzea.
- SOCKS protokoloa onartzen duten SSH bezeroekin zifratutako proxy konexioak sortuz sarea modu seguruan nabigatzea.
- SSHFS erabiliz urruneko zerbitzari batean dagoen katalogo bat modu seguruan muntatzea.
- Aurreko teknikak erabiliz zerbitzarien urruneko ikuskatze eta administrazio automatizatua.

4.3.3 Arkitektura

SSH-2 protokoloak RFC 4251-n definitu bezala ondo banatutako geruzez osatutako arkitektura dauka:

- **Garraio geruza** (RFC 4253): Geruza honetan ematen dira hasierako gako trukea, zerbitzariaren autentifikazioa eta zifratzearen, konpresioaren eta osotasunaren egiaztatzearen zehaztapena. Goiko geruzarekiko harremanetarako interfaze bat dauka, 32768 biterainoko testu lauzko paketeen trukaketa ahalbideratzen duena. Geruza honek antolatzen du gakoen birtrukaketa ere, orokorrean gigabyte bat datu transferitu ondoren edo ordubete pasa ondoren.
- **Erabiltzailearen identifikazio geruza** (RFC 4252): Geruza hau arduratzen da bezeroa autentifikatzeaz eta hainbat autentifikazio metodo eskaintzen ditu. Autentifikazioa bezeroak zuzendutakoa da (*client-driven*), pasahitza eskatzen zaigunean SSH bezeroa da eskaera egiten duena, ez zerbitzaria. Zerbitzariak bezeroaren autentifikazio eskaerei erantzun besterik ez die egiten. Erabiltzailea autentifikatzeko metodo erabilienen artekoak dira ondokoak:
  - **Pasahitza**: pasahitz bidezko autentifikazioa eta pasahitza aldatzeko aukera eskaintzen ditu. Metodo hau ez dute programa guztiek erabiltzen.
  - **Gako publikoa**: gako publiko bidezko autentifikazio metodoa, orokorrean, DSA eta RSA gako pareak onartzen ditu gutxienez eta beste inplementazio batzuk X.509 ziurtagiriak ere onartzen dituzte.
  - **Teklatu-interaktiboa** (RFC 4256): erabilera anitzeko metodo honetan zerbitzariak informazio eskaera bat edo gehiago bidaltzen ditu eta bezeroak bistaratu ondoren erabiltzaileak sakatutako erantzunak bidaltzen ditu bueltan. S/Key eta SecurID moduko erabilera-bakarreko pasahitzak erabiltzea ahalbideratzen du. OpenSSH-ren zenbait konfiguraziotan erabiltzen da, ostalarien autentifikaziorako PAM erabiltzen denean. Kasu batzuetan pasahitz arrunta erabiltzen duten bezeroei konexioa galarazten die.
  - **GSSAPI autentifikazio metodoak**: SSH autentifikazioa Kerberos 5 edo NTLM bezalako kanpoko mekanismoen bidez gauzatzea ahalbideratzen dute, *single sign on* (saio hasiera bakunak???) eskaintzea. Aukera hau erakundeei zuzendutako SSH-ren inplementazio komertzialek eskaini ohi dute, hala ere, OpenSSH-k ere badu GSSAPI-rekin lan egiteko prestatutako inplementazio bat.

- **Konexio geruza** (RFC 4254): Geruza honek SSH zerbitzuentzako kanal kontzeptua, kanal eskaerak eta eskaera globalak definitzen ditu. SSH konexio bakarrak hainbat kanal ostatatu ditzake aldi berean, bakoitza bi aldeetara datuak transferitzen.

  Kanal mota estandarrak:

  - **shell:** terminaleko shell, SFTP eta exec eskaerentzat (SCP eskaerak barne).
  - **direct-tcpip**: bezerotik-zerbitzarirako birbidalitako konexioentzat.
  - **forwarded-tcpip**: zerbitzaritik-bezerora birbidalitako konexioentzat.
- **SSHFP DNS erregistroa** (RFC 4255): ostalariaren gako publiko hatz-markak eskaintzen ditu ostalariaren autentifikazioan laguntzeko.

Arkitektura ireki honek malgutasun handia eskaintzen du, SSH shell saio seguruez gain hainbat erabilera ahalbideratuz.

4.3.4 Fitxategien transferentzia segurua

4.3.4.1 Secure copy (SCP)

Secure Copy edo SCP makina lokalaren eta urruneko makina baten artean edo urruneko bi makinen artean fitxategiak transferitzeko modu bat da eta Secure Shell (SSH) protokoloan oinarritzen da. SCP RCP-ren ordezko segurua da

SCP protokoloa sareko protokolo bat da, BSD RCP protokoloan oinarritua eta sarean konektatutako makinen arteko fitxategien transferentzia ahalbideratzen du. SCP-k SSH erabiltzen du datu transferentziarako eta autentifikaziorako, sarean zehar bidalitako datuen autentifikazioa eta konfidentzialtasuna ziurtatuz.

SCP-k erabiltzen duen ataka lehenetsia TCP 22 da. RCP-rekin gertatzen den bezala ez dago protokoloa definitzen duen RFC-rik.

Bezerotik zerbitzarira edo zerbitzaritik bezerora fitxategiak transferitzeko erabili ahal izateaz gain urruneko bi makinen arteko fitxategi transferentzia ere ahalbideratzen du. Azken kasu honetan, SCP bezeroak SSH konexio bat abiarazten du makina lokalaren eta urruneko makinetako baten artean eta bertatik SCP konexio bidez lotzen ditu urruneko bi makinak. Modu honetan urruneko bi makinen arteko komunikazioa zuzena da eta makina lokalak ez du bitartekari bezala lan egiten.

4.3.4.2 SSH File Transfer Protocol (SFTP)

SSH File Transfer Protocol (SFTP) sareko protokolo bat da, modu seguruan fitxategietara sarbidea, fitxategi transferentzia eta fitxategi kudeaketarako aukera eskaintzen dituena. *Internet Engineering Task Force* (IETF) erakundeak diseinatu zuen SSH 2.0 bertsioaren hedapen bezala, baina helburua beste protokoloekin ere erabili ahal izatea zen.

SCP protokoloak fitxategien eta katalogoen transferentzia bakarrik onartzen du, SFTP-k ordea urruneko fitxategiekin hainbat eragiketa egitea ahalbideratzen du, urruneko fitxategi sistema protokolo moduko zerbait baita. SCP-k ez bezala urruneko fitxategien transferentziak pausatu eta berrabiaraz ditzake, katalogoen edukiak zerrendatu eta fitxategiak ezabatu.

SFTP ez da FTP SSH-ren gainean, baizik eta hutsetik abiatuta berriz diseinatutako protokolo berri bat, FTP-ren ordezko seguru izateko sortua. Ez da nahastu behar FTPS-rekin ere (*FTP Secure* edo FTP-SSL).

Protokoloak berak ez du autentifikazio eta segurtasunik eskaintzen eta protokolo seguru baten gainean erabiliko dela suposatzen du.

4.3.5 Inplementazio librea: OpenSSH

OpenSSH sarean zehar zifraturiko konexioak ezartzea ahalbideratzen duen aplikazio bilduma librea da. Ondoko tresnak eskaintzen ditu:

- **ssh bezeroa**: Urruneko makina batetara shell sarbidea eskaintzen du. rlogin, rsh eta telnet-en ordezko segurua.
- **sshd zerbitzaria**: Zerbitzarirako SSH daemona.
- **scp**: rcp-ren ordezko segurua.
- **sftp**: ftp-ren ordezko segurua.
- **ssh-keygen**: Autentifikazio prozesuan erabiltzen diren RSA eta DSA gakoak sortzeko eta aztertzeko tresna.
- **ssh-agent** eta **ssh-add**: Autentifikazioa errazten duten tresnak. Gakoak prest edukitzen dituzte, aldioro pasa-esaldiak idatzi beharra saihestuz.
- **ssh-keyscan**: Ostalari zerrenda bat eskaneatu eta gako publikoak biltzen dituen tresna.

BSD lizentziapean kaleratutako software librea da.

Webgunea: http://www.openssh.com/

Iturburu kodea: http://www.openssh.org/portable.html

4.3.6 Bibliografia

- http://en.wikipedia.org/wiki/Secure_Shell
- http://en.wikipedia.org/wiki/Secure_copy
- http://en.wikipedia.org/wiki/SSH_file_transfer_protocol
- http://en.wikipedia.org/wiki/OpenSSH

4.4 Virtual Network Computing (VNC)

RFB protokoloa erabiltzen duen urruneko administrazio grafikoko sistema bat da. Teklatuaren eta saguaren informazioa transmititzen du urruneko ordenagailura, eguneratutako ingurune grafikoa itzuliz sarean zehar.

VNC erabiliz sistema eragile desberdinen artean konekta daiteke eta bezero bat baino gehiago konekta daitezke zerbitzari batera aldi berean.

Erabilera arrunten artean daude urruneko euskarri teknikoa eta konputagailu bateko fitxategiak eskuratu ahal izatea urrunetik, adibidez, bulegoko ordenagailuan dagoen fitxategi bat eskuratzea etxeko ordenagailutik edo mugikorretik.

4.4.1 Historia

VNC-ren jatorrizko garapena Erresuma Batuko *Olivetti & Oracle Research Laboratory*-n (ORL) egin zuten, Cambridge-n. 1999an AT&T-k erosi zuen laborategia eta 2002an itxi zuten. Garapen taldeko hainbat partaidek RealVNC sortu zuten VNC software libre eta komertzialaren garapenarekin jarraitzeko, euren artean zeuden Tristan Richardson (VNC-ren asmatzailea), Andy Harter (proiektu burua itxiera aurretik), James Weatherall, eta abar.

Jatorrizko iturburu kodea eta hainbat eratorri libreak dira (GNU GPL lizentzia). RFB protokoloa eranskinak onartzeko asmoz prestatua izan zenez ez dago bateragarritasun arazorik produktu eratorri hauen artean, VNC zerbitzari eta bezeroek abildadeak negoziatzen baitituzte diosalean (*handshaking*).

4.4.2 Oinarriak

VNC sistema batek 3 osagai ditu:

- **VNC zerbitzaria**
- **VNC bezeroa**
- **VNC protokoloa (RFB)**

Bezeroa zerbitzariko ataka batetara konektatzen da (lehenetsia 5900 da). Bezeroa web-arakatzaile bat erabiliz ere konekta daiteke, kasu honetan ataka lehenetsia 5800 delarik.

Zerbitzariak pantailaren framebufferreko laukizuzenak bidaltzen dizkio bezeroari, honek irudia berrosa dezan. Prozesu honek banda-zabalera handia behar izaten du eta hainbat metodo garatu dira beharrezko banda-zabalera gutxitzeko, adibidez kodeketa teknikak erabiltzea. Bezeroak eta zerbitzariak ze kodeketa mota erabiliko duten negoziatu behar dute lehenik eta behin. Kodeketa mota sinpleen eta bezero/zerbitzari guztiek onartzen dutenean lehen bidalketan pantailako pixel guztiak transmititzen dira (ezkerretik eskuinera eta goitik behera) eta hortik aurrera aldatzen diren pixelak bakarrik bidaltzen dira. Kodeketa mota honek oso ondo funtzionatzen du pantailaren zati txiki bat bakarrik aldatzen denean, adibidez, erakuslea mugitzean edo testua idaztean. Baina banda-zabalera handia behar du pantaila osoko pixelak aldatzen direnean.

VNC-rako ataka lehenetsia TCP 5900+N da, non N pantaila zenbakia den (N=0 pantaila fisikoaren kasuan). Zenbait inplementaziok HTTP zerbitzari bat ere abiarazten dute 5800+N atakan, Java applet bidezko VNC bezeroa eskaintzeko Java onartzen duten web nabigatzaileetan. Beste ataka zenbaki batzuk ere erabil daitezke beti ere bezeroa eta zerbitzaria ados jartzen badira.

Interneten zeharreko VNC-k ondo funtzionatzen du erabiltzaileak bi muturretan banda zabaleko konexioa badu, baina kasu batzuetan beharrezkoa izango da NAT, suhesi eta routerraren

konfigurazio aurreratuak erabiltzea, adibidez ataka birbidalketa. Erabiltzaile batzuk Hamachi bezalako Virtual Private Network (VPN) aplikazioak erabili nahi izaten dituzte gauzak errazteko. Beste aukera bat VNC konexioa LAN konexio moduan ezartzea da, VPN proxy bezala erabiliz.

VNC zerbitzaria martxan dagoen makinak ez du pantaila fisikorik eduki beharrik. Aplikazioak Xvnc bezala ager daitezke, X ikuskapen arrunt bat balitz bezala, baina konektatutako edozein VNC bezerotan bistaratuko dira pantaila fisikoan bistaratu ordez.

Gainera, VNC bidez zerbitzatutako pantaila eta zerbitzariaren pantailan bistaratzen dena desberdinak izan daitezke. Aldibereko hainbat X11 saio onartzen dituzten Unix/Linux konputagailuetan, horietako edozein saio zerbitzatzeko konfigura daiteke VNC edo saio berri propio bat hasteko nahi izanez gero. Posible da ordenagailu berdinetik hainbat VNC saio zerbitzatzea aldi berean. Microsoft Windows-en dagoenean zerbitzaria, erabiltzailearen uneko saioa da VNC bidez zerbitza daitekeen bakarra.

4.4.3 Segurtasuna

VNC ez da protokolo segurua. Pasahitzak testu lau bezala bidaltzen ez badira ere, irakurtzea lor daiteke zifratze gakoa eta zifratutako pasahitza, biak, saretik usnatzea lortuz gero. Hori dela eta, gutxienez 8 karaktereko pasahitzak erabiltzea gomendatzen da. Hala ere, zenbait inplementaziok 8 karaktereko muga ezartzen dute eta pasahitz luzeagoak erabiliz gero karaktere gehigarriak ez dituzte kontutan hartzen.

Segurtasun maila hobetzeko SSH edo VPN konexioen bidezko tunelak erabiltzea komeni da.

4.4.4 Mugak

VNC 3.x eta bertsio zaharragoek ez dute Unicode-ren erabilera onartzen, Latin-1 karaktere multzoa bakarrik transferi daitekeelarik arbeletik.

VNC protokoloa pixeletan oinarritzen da. Honek malgutasun handia ematen dion arren, ez da X11 edo Windows Remote Desktop Protocol bezain eraginkorra. Protokolo hauek grafikoen jatorrizkoak edo maila altuko komandoak (adibidez: “ireki leihoa”) bidaltzen dituzte, RFB-k pixel datu gordinak bidaltzen dituen bitartean.

4.4.5 RFB protokoloa

4.4.5.1 Sarrera

RFB protokoloa (*Remote FrameBuffer*, urruneko framebufferra) interfaze grafikoa urrunetik erabiltzea ahalbideratzen duen protokolo sinple bat da, *Virtual Networking Computing* (VNC) eta eratorrietan erabiltzen dena. *Framebuffer* mailan lan egiten duenez edozein leiho-sistematan erabil daiteke, adibidez, X11, Windows eta MacOS-en.

RFB benetan bezero soila erabiltzen duen protokoloa da, hau da, protokoloa diseinatzerakoan bezero aldetik ahalik eta baldintza gutxien jartzen saiatu ziren.

Protokoloak bezeroa egoeragabe egiten du. Bezeroa zerbitzari jakin batetik deskonektatu eta berriz konektatzen bada erabiltzaile-interfazearen egoera mantentzen da. Are gehiago, beste bezero bat erabil daiteke RFB zerbitzari berdinera konektatzeko. Bezero berriak bezero zaharrak ikusten duen erabiltzaile interfaze bera ikusiko du.

4.4.5.2 Ikuskatze protokoloa

Protokoloaren ikuskatze alderdiaren oinarria oso sinplea da eta esaldi honetan laburbil daiteke: “jarri pixelez osatutako laukizuzen bat emandako x,y posizioan”. Lehen begiradan oso eraginkorra ez dirudien arren, malgutasun handia eskaintzen du konexioaren banda zabalera, bezeroaren marrazte abiadura eta zerbitzariaren prozesatze abiadura bezalako parametro aldakorrekin lan egitean.

Laukizuzen hauen sekuentzia batek “*framebuffer* eguneraketa” bat osatzen du. Eguneraketa bat baliozko *framebuffer* egoera batetik beste baterako aldaketa da eta nolabait bideo baten fotograma baten moduko zerbait da. Eguneraketa bateko laukizuzenek orokorrean ez dute loturarik izaten euren artean baina ez da beti hala gertatzen.

Eguneraketa protokoloa bezeroaren eskariz jartzen da martxan. Hau da, bezeroak esplizituki eguneraketa eskaera bat egin ondoren soilik erantzuten du zerbitzariak. Honek protokoloari egoera desberdinetara egokitzeko gaitasun handia ematen dio, bezeroa edota sareko konexioa gero eta mantsoagoak izan eguneraketen maiztasuna ere neurri berean urrituko baita.

4.4.5.3 Sarrera protokoloa

Sarrera protokoloa teklatu eta botoi-aniztun erakusle aparailuan oinarritzen da. Sarrerako gertaera bat dagoen bakoitzean, hau da, erabiltzaileak teklatuko tekla bat edo erakusleko botoi bat sakatzen duen edo erakuslea mugitzen duen bakoitzean bezeroak zerbitzariari honen berri bidaltzen dio. Sarrerako gertaera hauek I/O gailu ez-estandarren bidez eman daitezke, adibidez, idazkera ezagutzen duen gailu bat erabiltzea teklatu bat erabiltzearen baliokidea da.

4.4.5.4 Pixel datuen irudikapena

RFB bezero batek eta zerbitzariak lehen aldiz konektatzean egiten duten lehen gauza pixel datuen formatua eta kodetze mota negoziatzea da. Negoziaketa honetan helburu nagusia bezeroarentzat gauzak ahalik eta errazen izatea da.

Pixel formatuak pixel bakoitzaren kolorearen errepresentazioak dira. Hauek dira pixel formatu ohizkoenak:

- **24-biteko edo 16-biteko “benetako kolore” formatuak**, non pixel bakoitza gorri, berde eta urdinaren intentsitateen konbinazio bezala adierazten den.
- **8-biteko “kolore mapak”**, non mapeatze arbitrarioak erabil daitezkeen pixelen balioak RGB intentsitateetara itzultzeko.

Pixelen kodetzea pixelen datuak sarean zehar bidaltzeko moduari dagokio. Pixelen datuen laukizuzen bakoitzak laukizuzenaren X,Y posizioak, zabalera eta luzera eta kodeketa mota adierazten dituen goiburuko bat darama eta ondoren adierazitako kodeketa erabiliz kodetutako datuak.

Ondoko kodeketa motak onartzen dira protokoloaren 3.8 bertsioan: Raw, CopyRect, RRE, Hextile eta ZRLE. Gehien erabiltzen direnak ZRLE, Hextile eta CopyRect dira ohiko mahaigainentzat konpresio onena eskaintzen dutenak baitira.

4.4.5.5 Protokoloaren hedapenak

- **Kodeketa mota berriak**: Protokoloari kodeketa mota berriak gehi dakizkioke nahiko modu errazean bertsio zaharragoekin bateragarritasuna mantenduz. Zerbitzari zaharragoek ez dituzte kontutan hartzen onartzen ez dituzten kodeketa berrien eskaerak eta negoziaketan

kodeketa zaharrago baten alde egiten dute. Bezero zaharrek ez dute ezagutzen ez duten kodeketa berririk eskatuko.

- **Sasi-kodeketak**: Bezero batek sasi-kodeketa baten eskaera egin diezaioke zerbitzariari protokoloaren hedapen jakin bat ezagutzen duela adierazteko. Hedapena ezagutzen ez duen zerbitzari batek ezikusia egingo dio. Hau da, bezeroak zerbitzariak hedapena ez duela ezagutzen suposatu behar du zerbitzaritik baieztapenik jaso ezean.
- **Segurtasun mota berriak**

Protokoloaren arduradunek, RealVNC Ltd.-ek, izendatutako protokolo zenbakiak errespetatu beharra dago. Bertsio zenbaki desberdin bat erabiliz gero gure aplikazioa ez da RFB/VNC-rekin bateragarria izango. Bateragarritasuna ziurtatzeko RealVNC Ltd.-ekin kontaktuan jartzea komeni da, gure kudeaketa mota eta segurtasun mota berriek arazorik ez dutela sortzen egiaztatzeko. Behar izanez gero http://www.realvnc.com -en aurki daiteke kontaktatzeko moduari buruzko informazio gehiago; VNC-ren posta-zerrendara mezu bat bidaltzea da kontaktatzeko biderik egokiena eta bide batez VNC komunitateak gure asmoen berri izango du.

4.4.5.6 Protokolo mezuak

RFB protokoloa TCP/IP konexio baten gainean erabili ohi da.

Hiru ataletan banatzen da:

- **Esku-emate fasea** (*Handshake phase*): Fase honen helburua erabiliko diren protokolo zenbakia eta segurtasun mota zehaztea da.

  Esku-ematea hasteko zerbitzariak bezeroari ProtocolVersion mezua bidaltzen dio. Honi esker, zerbitzariak ezagutzen duen protokolo bertsio zenbaki altuena lortzen du bezeroak.

  Ondoren, bezeroak erabiliko den protokolo bertsio zenbakia bidaltzen dio zerbitzariari. Bertsio zenbaki honek ez du zertan zerbitzariak bidalitako berdina izan behar, baina ezin du izan zerbitzariak ezagutzen duen protokolo berriena baino berriagoa. Atzeranzko bateragarritasuna eskaintzea da honen helburua.

  Momentu honetan existitzen diren protokolo zenbaki bakarrak 3.3, 3.7 eta 3.8 dira

  Protokolo bertsioa zehaztu ondoren, erabiliko den segurtasun mota zehazten da.

- **Abiarazpen fasea** (*Initialisation phase*): Fase honetan bezeroak ClientInit mezu bat bidaltzen dio zerbitzariari eta honek ServerInit mezu bat itzultzen dio.

  ClientInit mezuan bezeroak zerbitzariari jakinarazten dio aurretik konektatuta dauden bezeroen konexioak mantendu behar dituen edo konexioa eten behar dien, bezero honi sarbide esklusiboa emateko.

  ClientInit mezua jaso ondoren, zerbitzariak ServerInit mezua bidaltzen dio bezeroari. Mezu honetan zerbitzariaren framebufferraren altuera eta zabalera, pixel formatua (pixelak biteko, kolore sakonera, “benetako koloreak” erabili edo ez) eta mahaigainaren loturiko izena zehazten dira. Bezeroak aurrerago besterik zehaztu ezean pixel formatu hau erabiliko da.

- **Protokoloaren elkarrekintza fase normala** (*Normal protocol interaction*): Bezeroak nahi dituen mezuak bidaltzen ditu eta zerbitzariak beharrezko erantzunak ematen dizkio bueltan. Mezu hauek guztiak mezu-mota zehazten duen bytearekin hasten dira, jarraian mezuari dagozkion datuak doazelarik.

Bezeroak zerbitzariari bidaltzen dizkion ohiko mezuak honakoak dira:

- **SetPixelFormat**: FramebufferUpdate mezuetan erabiliko den pixel formatua zehazten du. Zerbitzariak ServerInit mezuan ezarritako formatua erabiltzerik ez duenean bakarrik erabiltzen da.
- **SetEncodings**: Bezeroak hobesten dituen pixel datuen kodeketa edota sasi-kodeketa motak zehazten ditu mezu honetan. Zerbitzariak hobespen hauei jaramon egingo die edo ez.
- **FramebufferUpdateRequest**: Bezeroak x-posizioa, y-posizioa, zabalera eta altuera balioek zehazturiko *framebufferra* nahi duela jakinarazteko erabiltzen da. Zerbitzariak orokorrean FramebufferUpdate mezu bat bidaliz erantzuten du. Hala ere, kasu batzuetan zerbitzariak mezu bakarrekin erantzungo die bezeroaren hainbat eskaerei.

  Framebuffer eguneraketa hauek inkrementalak izan ohi diren arren, bezeroak mahaigain osoa bidaltzeko eskaera ere egin dezake.
- **KeyEvent**: Zerbitzariari bezeroko teklatuan tekla bat sakatu dela adierazteko erabiltzen da. Tekla konbinazioak (Shift, Ctrl, Alt + tekla) bidaltzeko modua eskaintzen du.
- **PointerEvent**: Erakuslearen mugimendua edo erakuslearen botoi bat sakatu dela jakinarazteko erabiltzen da.
- **ClientCutText**: Bezeroak bere mozketa *buffer*rean testu berria daukala adierazten du. Momentuz Latin-1 karaktereekin bakarrik funtzionatzen du.

Zerbitzariak bezeroari bidaltzen dizkion ohiko mezuak berriz ondokoak dira:

- **FramebufferUpdate**: *Framebuffer* eguneraketa bat pixel datuen laukizuzen sekuentzia bat izaten da eta bezeroak egindako FramebufferUpdateRequest eskaera bati erantzunez bidaltzen da.
- **SetColourMapEntries**: Pixel formatuak kolore mapa bat erabiltzen duenean, zehazturiko pixel balioak emandako RGB intentsitateetara mapeatu behar direla adierazten du mezu honek.
- **Bell**: Bezeroak kanpaia edo txirrina jo behar duela adierazten du.
- **ServerCutText**: Zerbitzariak bere mozketa *buffer*rean testu berria daukala adierazten du. Momentuz Latin-1 karaktereekin bakarrik funtzionatzen du.

Mezu mota gehiago ere badauden arren, mezu ez-estandar bat bidali aurretik beste aldeak hedapen hori onartzen duela ziurtatu beharra dago.

4.4.6 Inplementazio libreak

4.4.6.1 Zerbitzariak

4.4.6.1.1 Vino

Vino GNOME mahaigainerako VNC zerbitzari lehenetsia da. Uneko mahaigaina beste ordenagailu batetara esportatzeko aukera eskaintzen du, urruneko erabilerarako edo euskarri teknikorako.

Software librea da. Kode biltegia: http://svn.gnome.org/svn/vino/trunk/

4.4.6.1.2 x11vnc

x11vnc VNC zerbitzari bat da.

Besteak beste ondoko ezaugarriak ditu:

- SSL/TLS zifraketa eta 2048 bit RSA autentifikazioa
- UNIX kontu eta pasahitzentzako euskarria
- Zerbitzari aldeko eskalatzea
- Portu bakarreko HTTPS/HTTP eta VNC
- TightVNC eta UltraVNC fitxategi transferentzia
- IPv6 euskarri osoa

GPL lizentziadun software librea da.

Webgunea: http://www.karlrunge.com/x11vnc/

Kode biltegia: http://sourceforge.net/projects/libvncserver/files/x11vnc/

4.4.6.2 Bezeroak

4.4.6.2.1 Vinagre: Urruneko Mahaigainen Ikustailea

Vinagre GNOME mahaigainerako VNC bezero lehenetsia da.

Ezaugarriak:

- Aldi berean hainbat makinetara konektatzeko aukera. Mahaigain bakoitza fitxa batean agertzen da, batetik bestera pasatzea erraztuz.
- Laster markak erabil daitezke, konexioak berriz konfiguratu beharrik ez izateko.
- Pasahitzak gogoratzeko aukera.
- VNC zerbitzariak aurkitzeko sare lokala nabigatzeko aukera eskaintzen du.
- VNC eta SSH protokoloak onartzen ditu. Etorkizunean Windows makinetara RDP erabiliz konektatzeko aukera gehitu nahi dute.
- SSH birbidalketak egiteko aukera.
- Pantaila irudiak egiteko aukera.

GPL lizentziapean kaleratuko software librea da.

Webgunea: http://projects.gnome.org/vinagre/

4.4.7 Bibliografia

VNC

- http://en.wikipedia.org/wiki/Vnc

RFB protokoloa

- RFB 3.8 Protokolo Estandarra: http://www.realvnc.com/docs/rfbproto.pdf

Vinagre

- http://en.wikipedia.org/wiki/Vinagre

x11vnc

- http://www.karlrunge.com/x11vnc/

4.5 NX

NX teknologia urruneko X Leiho Sistema konexioak modu azkar batean erabiltzea ahalbideratzen duen teknika da. Honi esker, bezeroak Linux edo Unix mahaigainetara konekta daitezke urrunetik, baita modem bidezko konexio mantsoekin ere. X11 protokoloa zuzenean konprimitzen duenez VNC baino eraginkorragoa da. Informazioa SSH bidez bidaltzen da, modu honetan bezero eta zerbitzari arteko informazio truke osoa zifratua delarik.

4.5.1 Historia

Italiako NoMachine enpresako Gian Filippo Pinzari-k sortu zuen, DXPC (*Differential X Protocol Compressor*) proiektuan oinarrituz.

2010eko abenduaren 21ean NoMachine-k iragarri zuen 4.0 bertsioa jabedun software bezala kaleratuko zutela. Ordura arte GNU *General Public License* erabili zuten NX teknologiaren zati handi batentzat, enpresentzat garapen komertzial ez-libreak ere eskaintzen zituzten bitartean.

Egoera berriari aurre egin eta aukera libre bat eskaintzeko FreeNX proiektua sortu zen.

2009ko uztailaren 7an, Google-k lizentzia libreko NX zerbitzaria iragarri zuen, Neatx.

4.5.2 Oinarriak

NX-k X11 datuak konprimitzen ditu sarean zehar bidali beharreko datu kopurua gutxitzeko.

NX-en bi osagai nagusiak **nxproxy** eta **nxagent** dira:

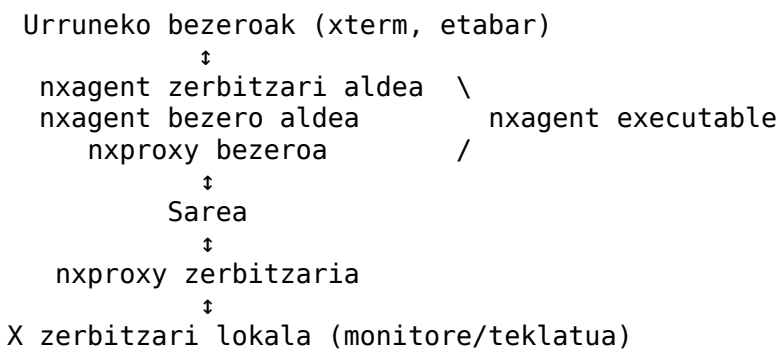
- **nxproxy** dxpc-ren eratorri bat da eta urruneko makinan (bezeroa X terminologian) eta makina lokalean (zerbitzaria X terminologian), bietan, abiarazten da. X zerbitzari bat simulatzen du bezeroan eta urruneko X protokolo eskaerak birbidaltzen ditu X zerbitzari lokalera.

  Ohiko eskema ondokoa da:

```
urruneko bezeroak (xterm, eta abar)
            ↕
      nxproxy bezeroa
            ↕
          Sarea
            ↕
      nxproxy zerbitzaria
            ↕
X zerbitzari lokala (monitore/teklatua)
```

  nproxy-k 1:10 eta 1:1000 arteko konpresio erlazioak lortzen ditu, beharrezko banda zabalera nabarmen gutxituz, baina ez ditu X-ren bidai sinkrono zirkular gehienak ezabatzen eta azken hauek dira, batez ere, X-ren latentziaren arduradun nagusiak.

- **nxagent** aldiz Xnest-en eratorria da eta urruneko (bezero) makinan abiarazten da orokorrean, X11 protokoloaren bidai zirkular gehienak saihestuz. nxproxy nxagent barruan integratzen da eta bien artean eraginkortasun ona lortzen dute banda zabalera txikiko/latentzia handiko sareetan.



X11-ren inplementazio funtzional bat duten sistemetan, nxproxy eta nxagent dira beharrezko osagai bakarrak urruneko X bezero eta X zerbitzari lokal bat modu arin batean lotzeko.

Hasiera batean X11 saioentzat diseinatua izan bazen ere, NX zerbitzaria proxy zerbitzari bezala erabil daiteke tunelak egiteko Remote Desktop Protocol (*Windows Remote Desktop Services* saioak) eta Virtual Network Computing saioetara, protokolo hauei aipatu ditugun abiadura hobekuntzak eskainiz.

4.5.3 Inplementazio libreak

4.5.3.1 Zerbitzariak

4.5.3.1.1 FreeNX

FreeNX NoMachine-ren NX zerbitzariaren inplementazio librea da. 2010eko abenduaren 21ean NoMachine-k 4.0 bertsioa jabedun software bezala kaleratuko zutela iragarri zuenean sortu zen proiektua. Garapena eta mantenuaren arduraduna Fabian Franz da.

GPL lizentziadun software librea da.

Webgunea: http://freenx.berlios.de/

http://developer.berlios.de/projects/freenx/

4.5.3.1.2 Neatx

Neatx Google-k sortutako NX zerbitzari librea da, NoMachine-ren NX teknologian oinarritua.

Garapena hastea erabakitzeko arrazoia existitzen ziren inplementazioak jabedun lizentziadunak edo mantentzen zailak zirela izan zen. NoMachine-ren inplementazioak ez dira guztiz libreak eta FreeNX librea da baina Googleren iritzitan mantentzen zaila den Bash, Expect eta C-ren nahasketa bat erabiliz programatuta dago. Hori dela eta, garapen propio bat egiten hasi ziren, helburu bezala malgutasuna eta erraz mantentzeko modukoa izatea izanik. Ia erabat Python erabiliz idatzita dago, salbuespenak BASH-eko script batzuk eta C-n idatzitako programa bat direlarik.

GPLv2 lizentziadun software librea da.

Kode biltegia: http://code.google.com/p/neatx/

PPA: https://launchpad.net/~freenx-team/+archive/ppa

4.5.3.2 Bezeroak

4.5.3.2.1 QTNX

QTNX QT liburutegia erabiltzen duen NX bezero bat da. Eseki eta berriz abiarazteko aukera ematen du.

Software librea da.

Kode biltegia: https://code.launchpad.net/ubuntu/+source/qtnx

4.5.3.2.2 OpenNX

OpenNX NX bezero libre bat da. Eseki eta berriz abiarazteko aukera ematen du.

LGPL lizentziadun software librea da.

Webgunea: http://opennx.net/

Kode biltegia: http://sourceforge.net/projects/opennx/

4.5.4 Bibliografia

NX

- http://en.wikipedia.org/wiki/NX_technology
- 4.0 bertsioaren aurkezpena:

  http://www.nomachine.com/news-read.php?idnews=331
- http://www.nomachine.com/documents/getting-started.php
- http://www.nomachine.com/documents/intr-technology.php

FreeNX

- https://help.ubuntu.com/community/FreeNX
- http://www.linux-magazine.com/Online/Blogs/Productivity-Sauce/Access-Your-Apps-and-Docs-Remotely-with-FreeNX

Neatx

- http://www.techworld.com.au/article/310857/google_releases_open_source_nx_server/
- http://google-opensource.blogspot.com/2009/07/releasing-neatx-open-source-nx-servier.html

4.6 Android

4.6.1 Android Sistema Eragilea

Android sistema eragilea smartphone eta tablet-en moduko gailu mugikorrentzako sistema eragile bat da. Googlek gidatutako Open Handset Alliance-k garatzen du. Open Handset Alliance hardware, software eta telekomunikazioetako 80 enpresaren arteko elkarlanetik sortutako aliantza da. Enpresa hauen artean daude, adibidez, Google, HTC, Sony, Dell, Intel, Motorola, Texas Instruments, Samsung, LG, T-Mobile eta Nvidia. Beren helburua gailu mugikorretarako estandar irekiak bultzatzea da.

Androiden oinarrian Linux nukleo bat dago. Hala ere, sistema eragile honen bereizgarri nabarienetako bat Dalvik makina birtualaren erabilera da. Sistema eragilea jarraian sakonago aztertuko dugu arren, bere konplexutasunaren ideia bat egiteko, 43 lengoaia desberdinetan idatzitako 12 milioi kode lerrok osatzen zuten Android sistema eragilearen banaketa (Linux kernela, adibideak eta abar barne) 2010eko maiatzaren 23an. Kontaketa CLOC (Count Lines Of Code) programa erabiliz egina dago (http://cloc.sourceforge.net/):

- 3,04 milioi lerro XML
- 2,82 milioi lerro C
- 2,08 milioi lerro Java
- 1,75 milioi lerro C++
- 1,15 milioi lerro C/C++ goiburu

Androiden iturburu kodearen zati nagusia software librea da, Apache lizentziaduna. Aurrerago aztertuko dugun bezala badira jabedun lizentzia erabiltzen duten osagaiak ere.

Komunitate zabal bat ari da Androiderako aplikazioak garatzen eta une honetan 250000 aplikazio baino gehiago daude eskuragarri.

4.6.2 Egitura (Software pila)

Linux nukleoa

Androidek Linux nukleoaren 2.6 bertsioa erabiltzen du segurtasuna, memoria eta prozesuen kudeaketa, sareko pila, energia kudeaketa eta kontrolatzaileak bezalako oinarrizko zerbitzuetarako. Hardwarearen eta nukleoaren gainean eraikitako software pilaren arteko abstrakzio geruza bezala jokatzen du.

Liburutegiak

Nukleoaren gaineko geruzan zenbait C/C++ liburutegi daude. Garatzaileek eskuragarri dituzte liburutegi hauek Android aplikazio *framework*aren bidez.

- **Sistemaren C liburutegia**: C sistema liburutegi estandarraren (libc) BSD-ko bertsioan oinarritutako inplementazioa, Linux-en oinarritutako txertatutako sistementzat egokitua.
- **Multimedia liburutegiak**: PacketVideo-ren OpenCORE-n oinarrituak, soinu eta bideo formatu ezagunenen erreprodukzio eta grabazioa ahalbideratzen dutenak, baita irudi fitxategi estatikoenak ere. Onartutako formatu batzuk: MPEG4, H.264, MP3, JPG eta PNG.
- **Gainazal kudeatzailea**: Bistaratze azpi-sistemarako sarrera kudeatzen du eta hainbat

aplikazioren 2D eta 3D geruza grafikoak osatzen ditu.

- **LibWebCore**: Web arakatzaile moderno baten nukleoa, Android nabigatzailea eta *web view* txertagarrian erabiltzen dena.
- **SGL**: 2D grafikoen motorra.
- **3D liburutegiak**: OpenGL ES 1.0 APIetan oinarritutako inplementazioa. Eskuragarri dagoenean 3D hardware azelerazioa erabiltzen da edo bestela biziki optimizatutako 3D software bilbatzaile edo rasterizatzailea.
- **FreeType**: Bitmap eta bektore letra-tipoen errendatzea (*rendering*).
- **SQLite**: Aplikazio guztiek eskuragarri duten datu-base erlazional ahaltsu eta arinaren motorra.

Android Runtime

Android GNU/Linux banaketa arrunt batetik desberdintzen duena *Android Runtime*-a da. Aplikazioentzako *framework*-aren oinarria osatzen du eta bi osagai ditu:

- **Oinarrizko liburutegiak (*Core libraries*)**

  Java programazio lengoaiaren oinarrizko liburutegietan eskuragarri dagoen funtzionalitatearen zati handi bat eskaintzen duten liburutegiak ditu Androidek.

- **Dalvik makina birtuala**

  Android aplikazio bakoitzak prozesu bereizi bat erabiltzen du, Dalvik makina birtualaren instantzia propioarekin. Hainbat makina birtual aldi berean modu eraginkorrean exekutatu daitezen prestatua izan da Dalvik. Dalvik makina birtualak Dalvik exekutagarri formatuko (.dex) fitxategiak exekutatzen ditu eta memoria erabilpen minimoarekin lan egiteko prestatua izan da. Makina birtuala erregistroetan oinarritzen da eta “dx” tresnaren bitartez .dex formatura bihurtutako Java lengoaian konpilatutako klaseak erabiltzen ditu.

  Dalvik makina birtualak Linux nukleoa erabiltzen du harilkatze eta maila baxuko memoria kudeaketarako.

Aplikazio *Framework*-a

Android-en jatorrizko aplikazioek erabiltzen duten *framework*-aren API berdinak erabil daitezke software garapenean. Aplikazioen arkitektura osagaien berrerabilpena errazteko diseinatua dago; edozein aplikaziok beste aplikazioen ahalmenak erabil ditzake, beti ere *framework*-ak eskaintzen dituen segurtasun neurriak errespetatuz. Modu berean, edozein osagai ordezka daiteke.

Aplikazio guztien azpian zerbitzu eta sistema multzo bat dago:

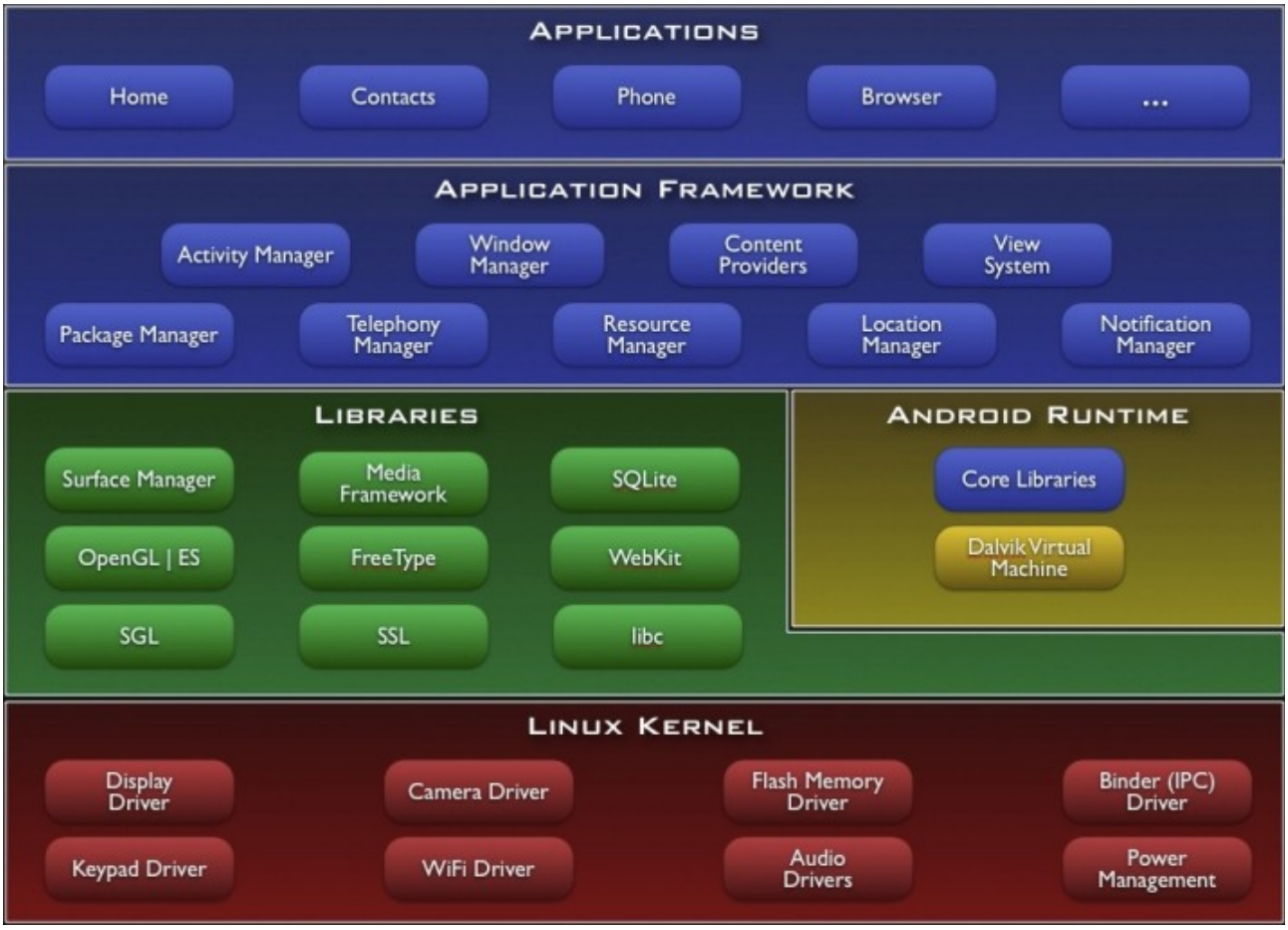
- ***View*** multzo zabal eta hedagarria, aplikazioak sortzeko erabil daitekeena. Horien artean daude zerrendak, saretak (*grid*-ak), testu kaxak, botoiak, web arakatzaile txertagarri bat, ...
- **Eduki hornitzaileak** (*Content providers*): Beste aplikazioen datuak eskuratu edo gure aplikazioaren datuak partekatu ahal izatea ahalbideratzen dute, hala nola kontaktuak.
- **Baliabide kudeatzailea** (*Resource manager*): Kodea ez diren baliabideetara sarbidea izatea ahalbideratzen du: testu kateak, grafikoak, diseinuak (*layouts*), ...
- **Jakinarazpen kudeatzailea** (*Notification manager*): Egoera barran abisu pertsonalizatuak

bistaratzea ahalbideratzen du.

- **Aktibitate kudeatzailea** (*Activity manager*): Aplikazioen bizi-zikloa kudeatzen du.

Aplikazioak

Androidekin batera Java programazio lengoaia erabiliz programatutako hainbat aplikazio datoz: SMS programa, egutegia, mapak, arakatzailea, kontaktuak eta abar.



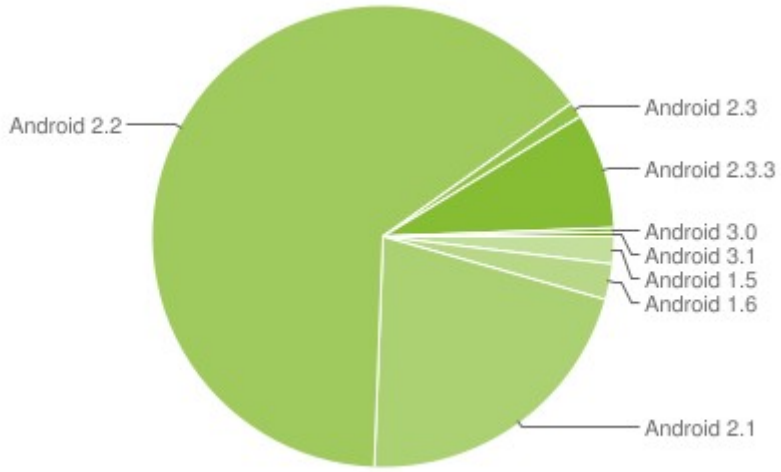
4.6.2.1 Bertsioak

- **1.5 (Cupcake)**
- **1.6 (Donut)**
- **2.0/2.1 (Eclair)**: Erabiltzaile interfazea berriztatzeaz gain HTML5, W3C Geolokalizazio API-a eta Exchange ActiveSync 2.5-ekin bateragarritasuna izan ziren berritasun nagusiak.
- **2.2 (Froyo)**: Eraginkortasun hobekuntzak JIT optimizazioari eta Chrome V8 JavaScript motorrari esker. *Wi-Fi hotspot tethering* eta Adobe Flash euskarriaren gehikuntza.
- **2.3 (Gingerbread)**: Erabiltzaile interfazearen birdefinizioa, pantailako teklatuaren eta kopiatu/itsatsi gaitasunen hobekuntza eta Near Field Communication-ekiko euskarria.
- **3.0/3.1 (Honeycomb)**: Tabletetara zuzendutako bertsioa, pantaila handientzat egokitua eta hainbat berrikuntza ekarri dituena erabiltzaile interfazean. Gainera, nukleo aniztun

prozesagailuak eta hardware bidez azeleratutako grafikoak onartzen ditu.

- **3.2 (Honeycomb)**: 3.1-en antzekoa baina 7 hazbeteko tabletentzat egokitua. 2011ko ekainaren 20an Huawei Technologies-ek MediaPad eman zuen ezagutzera, 3.2 Honeycomb erabiltzen duen munduko lehen 7 hazbeteko nukleo bikoitzeko tableta (1,2GHzko Qualcomm Chipset-a erabiltzen du).
- **4.0 (Ice Cream Sandwich)**: Gingerbread eta HoneyComb-en konbinazioa. 2011ko maiatzaren 10ean iragarri zuten Google I/O-n eta 2011ko azken hiruhilekoan kaleratuko da.

Ondoko grafikoan eta taulan ikus daiteke 2011ko ekainean bertsio bakoitzak zuen ehunekoa:



Bertsioa	API Maila	Portzentaia
Android 1.5	3	1.9%
Android 1.6	4	2.5%
Android 2.1	7	21.2%
Android 2.2	8	64.6%
Android 2.3 - Android 2.3.2	9	1.1%
Android 2.3.3 - Android 2.3.4	10	8.1%
Android 3.0	11	0.3%
Android 3.1	12	0.3%

4.6.3 Android %100 librea: Replicant proiektua

Replicant mugikorretarako sistema eragile bat da, Androiden oinarritua eta %100 software librea izatea helburu duena.

Replicant proiektua Software Freedom Conservancy (http://sfconservancy.org/) eta LibrePlanet Italia (http://groups.fsf.org/wiki/LibrePlanetItalia) elkarteen lankidetzari esker sortu zen.

Android-ek gehienbat Apache 2.0 lizentzia erabiltzen du eta Linux nukleoak GPLv2. Baina, lehenetsitako osagai askok jabedun lizentzia dute, hala nola Google Apps-ek (Gmail, GMaps, GTalk,...). Nabarmentzekoa da hardwarea zuzenean ukitzen duten osagai ia guztiek jabedun lizentzia dutela. Adibidez, HTC Dream-ek duen jabedun softwarearen artean daude:

- **Telefonoaren oinarrizko funtzionalitaterako beharrezko osagaiak**: GSM irrati interfaze liburutegia, oinarrizko audio liburutegia, audio mailekin erlazionatutako liburutegia, ...
- **Telefono funtzionalitatearekin zerikusirik ez duen hardwareak behar dituen osagaiak**: GPS liburutegia, pantaila eta teklatuaren atzeko argitasuna eta beste LEDekin erlazionatutako liburutegia, azelerometroa eta iparrorratza kudeatzeko liburutegia, sentsore daemona, *bluetooth*-aren *firmware*-a, kamera kontrolatzeko liburutegia, ...
- **Multimedia kodetzea eta deskodetzearekin erlazionatutako osagaiak**
- **Bestelakoak**: 3D errendatzea, Pinyin karaktere sarrera, Ahots-ezagutza, *Streaming Pocket Video Windows Media Decoder* liburutegia, ASF analizatzailea libpvasf-entzat, ...

Webgunea: http://replicant.us/

Iturburu kodea: http://gitorious.org/replicant

Posta zerrenda: http://lists.osuosl.org/mailman/listinfo/replicant

4.6.3.1.1 Euskarridun gailuak

- **HTC Dream / G1 / ADP1** (Une honetan HTC Dream da Replicant erabiltzeko aparailurik egokiena)
- **Google Nexus One**
- Euskarridun gailuen zerrenda eguneratua: http://replicant.us/supported-phones/

4.6.3.1.2 Google apps-entzako ordezko libreak

- **Android Market** -> **FDroid** (http://f-droid.org/)
- **Gmail** -> **K9 (**http://code.google.com/p/k9mail/, https://github.com/k9mail/k-9**)**
- **Google Maps** -> **OsmAnd** (Onena, https://code.google.com/p/osmand/), **gvSIG Mini Maps**, **OSMdroid**
- **Google Talk** -> **CSipsimple** (Onena, http://code.google.com/p/csipsimple/), **Beem** (http://www.beem-project.com/), **Gibber** (https://guardianproject.info/apps/gibber/), **Sipdroid?**, **Nimbuzz**
- **Youtube** -> Ez dago jabedun aplikazioen beharrik, nahikoa da youtube-ren mugikorretarako bertsioa erabiltzea (rstp-rekin dabil, ez adobe flash-ekin).
- **Genie Widget** (News/Weather widget bezala ere ezaguna) -> Albisteak: BBC News edo NPR News. Eguraldia: Ez dago jabedun aplikazioen beharrik, gustukoen duzun eguraldi webgunearen mugikorretarako bertsioa erabili.

4.6.4 Androiderako aplikazioen garapenerako tresnak

4.6.4.1 Android Software Garapen Tresna-jokoa *(Software Development Kit, SDK)*

Androiderako aplikazioen garapena errazteko prestatutako tresnen bilduma da. Tresna horien artean daude: araztailea, liburutegiak, emulatzailea, dokumentazioa, adibide kodea, tutorialak, ...

GNU/Linux, MacOS X (10.4.9 edo berriagoa) eta Windows (XP edo berriagoa) plataformetarako eskuragarri dago.

Eclipse software garapenerako ingurunearekin (IDE) integra daiteke *Android Development Tools* pluginari esker. Hala ere, nahi izanez gero edozein testu editore ere erabil daiteke Java eta XML fitxategien ediziorako eta komando lerroko tresnekin (*Java Development Kit* eta Apache Ant) sortu, eraiki eta aratz daitezke Androiderako aplikazioak.

4.6.4.2 Android SDK eta AVD kudeatzailea (Android SDK and AVD manager)

Androiden eta hirugarrenen (LG, Samsung, Sony Ericsson, ...) software biltegietatik SDK plataforma desberdinak, kode adibideak, dokumentazioa eta abar eskuratu eta instalatzea errazten du.

Horretaz gainera, Android gailu birtualen (Android Virtual Devices, AVD) kudeaketa errazten du. Hau da, tresna honen bidez, Android gailu birtualak sortu, editatu, ezabatu, konpondu eta abiaraz ditzakegu. Gailu birtualak software biltegietatik deskargatutako SDK plataformak erabiliz sortzen dira aurrerago ikusiko dugun bezala.

4.6.4.3 Android emulatzailea

QEMU-n oinarritutako aplikazio bat da Android emulatzailea eta Android aplikazioak egikaritzea ahalbideratzen duen ARM gailu mugikor birtual eskaintzen du. QEMU geruzak ARM makina kodea dinamikoki itzultzen du garapenean erabiltzen ari garen ordenagailuaren arkitekturara.

Hainbat hardware ezaugarrirentzako euskarria eskaintzen du, hala nola:

- ARMv5 CPUa eta dagokion memoria kudeaketa unitatea (*memory-management unit*, MMU)
- 16 biteko LCD pantaila
- Teklatu bat edo gehiago
- Soinu txip bat sarrera/irteera ahalmenekin
- Flash memoria partizioak (disko irudi fitxategi bezala emulatuak)
- GSM modema, SIM txartel simulatuarekin

Android SDK eta AVD kudeatzailea erabiliz abiaraz daiteke edo <android-SDK>/tools/ katalogoko *emulator* komandoa erabiliz. *emulator* tresnaren aukeren erreferentzia webgune honetan aurki daiteke: http://developer.android.com/guide/developing/devices/emulator.html#startup-options

ADT plugina erabiliz gero automatikoki abiaraz dezakegu Eclipse garapen ingurunetik.

4.6.4.4 Android gailu birtualak (Android Virtual Devices, AVD)

Android gailu birtual bat (AVD) Android emulatzailearen konfigurazio bat da, emulatutako gailu baten hardware eta software aukerak zehaztea ahalbideratzen duena. AVD bat sortzeko modurik

errazena Android SDK eta AVD kudeatzailea erabiltzea da baina komando lerrotik ere sor daiteke <android-SDK>/tools/ katalogoko *android* izeneko tresna erabiliz. *android* tresnaren aukeren erreferentzia webgune honetan aurki daiteke: http://developer.android.com/guide/developing/devices/managing-avds-cmdline.html

AVD batek ondoko osagaiak ditu:

- **Hardware profila**: Gailu birtualaren hardware ezaugarriak zehazten ditu. Adibidez, kamerarik duen edo ez, QWERTY teklatu fisiko bat duen edo ez, memoria kopurua, ...
- **Sistemaren irudi batetara mapeatzea**: Erabili nahi duzun Android plataformaren bertsioa zehatz daiteke. Bertsio estandar bat erabil daiteke (Jatorrizko Android) edo hirugarrenek prestatutakoa.
- **Bestelako aukerak**:
  - **Emulatzailearen azala** (*skin*): Pantailaren neurria, itxura eta abar.
  - **SD txartelaren ezaugarriak**
- **Biltegiratze eremua**: Gailuaren erabiltzaile-datuak (instalatuko aplikazioak, ezarpenak, ...) eta emulatutako SD txartelaren edukia gordetzeko.

4.6.4.5 Eclipserako Android Garapen Tresnak (Android Development Tools, ADT) plugina

Android Development Tools (ADT) Eclipse IDEarentzako plugin bat da, Androiderako aplikazioentzako garapen ingurune integratu eta ahaltsua eskaintzen duena.

Modu erraz batean Androiderako proiektu berriak eta aplikazioen erabiltzaile interfazeak sortu, *Android Framework* APIan oinarritutako osagaiak gehitu, XML fitxategiak editatu, Android SDK tresnak erabiliz aplikazioak araztu eta aplikazioa banatzeko .apk fitxategiak esportatzea ahalbideratzen du.

Androiderako proiektuak garatzen hastea asko errazten du eta guztiz gomendagarria da erabiltzea.

4.6.4.6 Android Native Development Tools (NDK)

Android NDK aplikazioetan kode natiboa (C, C++) erabiltzea ahalbideratzen duen SDK-ren gehigarri bat da. Honi esker C eta C++ erabiliz idatzitako liburutegiak berrerabil ditzakegu gure aplikazioetan. NDK-k liburutegi hauek gure aplikazio pakete fitxategietan (.apk) integratzen lagunduko digu.

Beharrezkoa denean bakarrik erabiltzea gomendatzen da, kasu batzuetan kode natiboa erabilita eraginkortasuna hobetuko bada ere aplikazioaren konplexutasuna handitzea ekarriko baitu. Kode natiboa erabiltzeko hautagai egokienak existitzen diren C/C++ liburutegien berrerabilpena eta memoria asko erabiltzen ez duten eta CPU-aren erabilera intentsiboa egiten duten operazioak dira, adibidez, seinale prozesatzea, simulazio fisikoak, ...

4.6.4.7 Android Debug Bridge (ADB)

Android emulatzaile batekin edo benetako gailu batekin komunikatzea ahalbideratzen duen komando lerroko tresna da. Bezero-zerbitzari motako programa bat da eta 3 osagai ditu:

- **Bezeroa**: Garapeneko ordenagailuan exekutatzen da. Komando lerrotik dei daiteke adb

komandoaren bidez. ADT pluginak eta DDMS-k ere adb bezeroak sortzen dituzte.

- **Zerbitzaria**: Garapeneko ordenagailuan atzeko planoan exekutatzen den prozesu bat da. Emulatzailean edo gailuan dagoen adb daemonaren eta bezeroaren arteko komunikazioa kudeatzen du.
- **Daemona**: Emulatzaile edo gailuan atzeko planoko prozesu bezala exekutatzen da.

adb tresna <sdk>/platform-tools/ katalogoan aurkitzen da.

adb bezero bat abiaraztean, adb zerbitzaririk martxan ba ote dagoen egiaztatzen du eta ez badago, bat abiarazten du. Zerbitzaria TCP 5037 portu lokalean entzuten hasten da abiaraztean eta adb bezeroak bidalitako mezuen zain gelditzen da, adb bezero eta zerbitzariak 5037 ataka erabiltzen baitute komunikatzeko.

Ondoren, zerbitzariak konexioak ezartzen ditu martxan dauden emulatzaile/gailu guztietara. Emulatzaile/gailuak aurkitzeko 5555 eta 5585 arteko portu bakoitiak eskaneatzen ditu, horiek baitira emulatzaile/gailuek erabiltzen dituztenak. Adb daemon bat aurkitzen duenean konexio bat ezartzen du portu horretara. Emulatzaile/gailu bakoitzak jarraian dauden bi portu erabiltzen ditu: portu bikoiti bat kontsola konexioetarako eta portu bakoiti bat adb konexioetarako. Adibidez:

Lehen emulatzailea, kontsola:5554

Lehen emulatzailea, adb:5555

Bigarren emulatzailea, kontsola:5556

Bigarren emulatzailea, adb:5557

Zerbitzariak instantzia guztiekin konexioak ezarri ondoren, adb komandoak erabil daitezke instantzia bakoitza kontrolatzeko (*script*-ak ere erabil daitezke). adb edo adb help exekutatuta eskuragarri dauden komandoen zerrenda bistaratzen da.

adb komandoen exekuzioa:

Komando lerrotik edo script bidez adb komandoak exekutatzeko hau da formatu orokorra:

```
adb [-d|-e|-s <serie-zenbakia>] <komandoa>
```

-d aukera: USB bidez konektatutako gailu bakarrari komandoak zuzentzeko erabiltzen da. Gailu bat baino gehiago badaude konektatuta errorea ematen du.

-e aukera: Martxan dagoen emulatzaile bakarrari komandoak zuzentzeko erabiltzen da. Emulatzaile bat baino gehiago badaude martxan errorea ematen du.

-s <serie-zenbakia>: Emulatzaile/gailu bat baino gehiago daudenean aldi berean konektatuta, zehaztutako serie-zenbakia duenari zuzentzen dizkio komandoak. Errorea ematen du ez bada serie-zenbakirik zehazten.

Ondoren komando arruntenak aztertuko ditugu:

Konektatutako emulatzaile/gailuen instantzien zerrenda lortzea:

Lanean hasi aurretik, konektatutako emulatzaile/gailuen instantziak ezagutzea komeni da, horretarako *devices* komandoa erabiltzen da.

```
adb devices
```

Konektatutako instantzia bakoitzeko lerro bat duen zerrenda bat itzuliko digu, ondoko

informazioarekin:

- **Serie-zenbakia**: adb-k emulatzaile/gailu instantzia bakoitza identifikatzeko sortzen duen katea da. Formatua: <mota>-<kontsola-ataka> Adibidez: emulator-5554
- **Egoera**: konexioaren egoera. 3 egoera posible daude:
  - **offline**: instantzia ez dago adb-ra konektatuta edo ez du erantzuten.
  - **device**: instantzia adb zerbitzarira konektatuta dago. Honek ez du esan nahi Android sistema behar bezala abiarazita edota erabilgarri dagoenik, instantzia adb-ra abiarazten ari den bitartean konektatzen baita. Hala ere, abiarazi ondoren, hau da erabilgarri dagoen instantzia baten egoera normala.
  - **no device**: adb-ra konektatutako emulatzaile/gailu instantziarik ez dago.

Adibidea:

```
adb devices
```

List of devices attached

emulator-5554 device

emulator-5556 device

emulator-5558 device

Komandoak emulatzaile/gailu instantzia jakin bati zuzentzea:

adb komandoak instantzia jakin bati zuzentzeko -s aukera erabiltzen da:

```
adb -s <serie-zenbakia> <komandoa>
```

Serie-zenbakia lortzeko lehen aipatutako devices komandoa erabiltzen da.

Adibidea:

```
adb -s emulator-5556 install helloWorld.apk
```

Kontutan izan -s aukera erabiltzean ez bada instantziaren serie-zenbakirik adierazten errore mezu bat agertzen dela

Aplikazio bat instalatzea:

Instantzia batetara aplikazioak kopiatu eta instalatzeko install komandoa erabiltzen da. Kontutan izan apk-ra iristeko bide izen erlatibo edo absolutua gehitu behar dela, ez dela nahikoa apk-ren fitxategi-izenarekin:

```
adb install [-r] [-s] <apk-rako_bidea>
```

-r : Aplikazioa berriz instalatu baina bere datuak mantendu

-s : SD txartelean instalatu aplikazioa (aukera lehenetsia gailuaren barneko memorian instalatzea da)

Eclipse IDE-a ADT plugin-arekin ez dago horrelakorik egin beharrik, IDE-a arduratzen baita apk fitxategia kopiatu, instalatu eta abiarazteaz.

Aplikazio bat desinstalatzea:

```
adb uninstall [-k] <aplikazio-paketea>
```

-k : aukera hau zehaztuz gero aplikazioaren datuak eta *cache* direktorioak ez dira ezabatzen

Portu birbidalketa:

Portu birbidalketak egiteko forward komandoa erabiltzen da. Adibidez, ostalariaren 6100 ataka emulatzaile/gailu-aren 7100 atakara birbidali nahi izanez gero:

```
adb forward tcp:6100 tcp:7100
```

UNIX domeinu *socket* izen abstraktuetara ere egin daitezke birbidalketak:

```
adb forward tcp:6100 local:logd
```

Fitxategiak kopiatzea emulatzaile/gailutik edo emulatzaile/gailura:

Fitxategiak kopiatzeko pull eta push komandoak erabiltzen dira. Emulatzaile/gailuan .apk fitxategiak kopiatu eta instalatzen dituen install komandoak ez bezala push eta pull komandoek edozein fitxategi mota edozein katalogotara kopia dezakete, ordenagailutik instantziara edo alderantziz.

Adibidea:

```
adb pull /sdcard/adibidea.txt /home/asier/adibidea.txt
```

Emulatzaile/gailutik ordenagailura fitxategi bat edo direktorio bat (errekurtsiboki) kopiatzeko pull erabiltzen da:

```
adb pull <instantziako_fitxategirako_bidea> <ordenagailuko_fitxategirako_bidea>
```

Ordenagailutik emulatzaile/gailura kopiatzeko berriz push erabiltzen da:

```
adb push <ordenagailuko_fitxategirako_bidea> <instantziako_fitxategirako_bidea>
```

Adibidea:

```
adb push /home/asier/adibidea.txt /sdcard/adibidea.txt
```

Script-etan erabiltzeko komandoak:

`adb get-serialno` : Uneko adb instantziaren serie-zenbaki katea inprimatzen du

`adb get-state` : Uneko egoera inprimatzen du. Hiru hauetako bat izaten da:

offline | *bootloader* | *device*

`adb wait-for-device` : Exekuzioa geldiarazten du gailua online egon arte, hau da, bere egoera device izan arte. Komando hau beste komandoekin batera erabil daiteke, komandoa exekutatzean gailua online dagoela ziurtatzeko.

Adibidez,

```
adb wait-for-device shell getprop
```

Oharra: wait-for-device komandoak ez du itxaroten sistema osoa guztiz abiarazi arte. Hori dela eta, kontuz ibili behar da gailua guztiz abiarazita behar duten komandoekin. Adibidez, install komandoak Android-en pakete-kudeatzailea behar du eta hau ez da eskuragarri egoten sistema guztiz abiarazi arte. Beraz, adb wait-for-device install <apk>.apk komandoak errorea emango luke, adb zerbitzarira konektatu orduko exekutatuko bailitzateke, gailua guztiz abiarazi aurretik.

Shell komandoen exekuzioa emulatzaile/gailuan:

Adb-k ash shell bat eskaintzen du emulatzaile edo gailuan komandoak exekutatzeko, emulatzaile/gailuaren adb urruneko shell erabiliz edo erabili gabe. Komandoen binarioak emulatzaile/gailuaren /system/bin/... direktorioan egon ohi dira.

Urruneko shell-a erabili gabe komando bakar bat exekutatzeko honela erabili behar da shell komandoa:

```
adb [-d|-e|-s {<serieZenbakia>}] shell <shellKomandoa>
```

Emulatzaile/gailu instantziaren urruneko shell-a erabiltzeko berriz:

```
adb [-d|-e|-s {<serieZenbakia>}] shell
```

Urruneko shell-a utzi nahi denean CTRL+D edo exit erabili behar da saioa bukatzeko.

sqlite3 datubaseak urruneko Shell bidez aztertzea:

Aurreko atalean aipatu moduan urruneko shell bidez instantzia batetara konektatu ondoren, komando-lerroko sqlite3 programa erabil daiteke Androideko aplikazioek sortutako SQLite datubaseak aztertzeko. sqlite3 tresnak hainbat komando interesgarri eskaintzen ditu, adibidez, .dump taula baten edukiak inprimatzeko eta .schema existitzen den taula baten CREATE sententzia inprimatzeko. SQLite komandoak nahierara exekutatzeko aukera ere ematen du, adibidez, taulak sortu, SQL sententziak exekutatu, eta abar.

sqlite3 komandoa exekutatzean datubaserako bide osoa zehatz daiteke. Datubaseak /data/data/<pakete_izena>/databases/ katalogoan egoten dira.

Adibide bat,

```
$ adb -s emulator-5554 shell
# sqlite3 /data/data/com.example.google.rss.rssexample/databases/rssitems.db
```

SQLite version 3.3.12

Enter ".help" for instructions

.... nahi diren komandoak exekutatu eta ondoren irten...

```
sqlite> .exit
```

.exit komandoa exekutatzean edo CTRL+D sakatzean sqlite3 utzi eta urruneko shell-era itzuliko gara.

Probak egiteko tximua (*UI/Application Exerciser Monkey*):

Gure simulatzaile/gailuan sasi-ausazko erabiltzaile-inputak (klikak, ukipenak, keinuak eta abar) eragiten dituen programa da Monkey. Garatzen ari garen aplikazioei errepikatzeko moduko ausazko estres testak egiteko erabiltzen da.

Aukera asko eskaintzen dituen arren honela laburbil daitezke:

- Oinarrizko konfigurazio aukerak, adibidez gertaera kopurua.
- Mugaketa operazionalak, adibidez testa aplikazio bakarrera mugatzea.
- Gertaera motak eta maiztasuna.
- Arazte aukerak

Monkey martxan dagoenean gertaerak sortu eta sistemari bidaltzen dizkio. Bien bitartean, sistema aztertzen du hiru zentzutan:

- Aplikazio pakete bat edo batzuetara mugatu badugu, beste paketeetara nabigatzeko saiakerak behatzen ditu eta blokeatu.
- Aplikazioak huts egiten badu edo kudeatu gabeko errore bat ematen bada, Monkey gelditu egiten da eta errorearen berri eman.
- Aplikazioak ez du erantzuten motako errore bat gertatzen bada, Monkey gelditu egiten da eta errorearen berri eman.

Monkey komando-lerrotik edo script batetik abiaraz daiteke. Emulatzaile/gailuan lan egiten duenez bertako shell batean abiarazi behar da. Horretarako aukera bat komando bakoitzari adb shell gehitzea da eta beste aukera emulatzaile/gailuan shell saio bat hasi eta bertan exekutatzea komandoak.

Oinarrizko sintaxia ondokoa da:

```
adb shell monkey [aukerak] <gertaera-kopurua>
```

Aukerarik zehazten ez bada Monkey modu isilean (ez-berritsuan) abiaraziko da eta helburuan instalatuta dauden pakete guztiei bidaliko dizkie gertaerak. Hala ere, orokorrean hainbat aukera zehaztu ohi dira. Adibidez, hurrengo komandoak gure aplikazioa soilik abiaraziko du, sasi-ausazko 500 gertaera bidaliko dizkio eta emaitzak modu berritsuan emango dizkigu:

```
adb shell monkey -v -p gure.paketearen.izena 500
```

Ikerketa honen eremutik kanpo geratzen denez gehiago sakonduko ez dudan arren Monkey komandoak aukera asko eskaintzen ditu, interesa izanez gero hurrengo helbidean aurki daiteke informazio gehiago: http://developer.android.com/guide/developing/tools/monkey.html#reference

Erabil daitezkeen adb shell komandoen zerrenda nola lortu:

Simulatzaile/gailuaren instantziak onartzen dituen onartzen dituen komandoen zerrenda lortzeko ondoko komandoa exekutatu behar da:

```
adb shell ls /system/bin
```

logcat komandoen erabilera:

logcat komandoa erabil daiteke sistemaren log bufferrak ikusi eta aztertzeko. Erabilera orokorra ondokoa da:

```
[adb] logcat [<aukera>] ... [<iragazki-zehaztapena>]
```

logcat komandoa garapeneko konputagailutik erabil daiteke edo urruneko adb shell bidez emulatzaile/gailu instantzian. Adibidez, log irteera garapeneko ordenagailuan ikusteko:

```
$ adb logcat
```

Urruneko adb shell bat erabiliz gero berriz:

```
# logcat
```

Log irteeraren iragazketa:

Androideko log mezu guztiek etiketa (tag) bat eta lehentasun bat dituzte eta outputaren lehen bi zutabeetan agertzen dira.

- Mezu baten etiketa testu kate labur bat izaten da, mezuaren jatorria adierazten duena, hau da, sistemaren ze osagaitan duen jatorria.
- Lehentasuna hurrengo karaktereetako bat izaten da, lehentasun txikitik handiagora ordenatuta:
  - V : Verbose (Berritsua, lehentasun txikiena)
  - D : Debug (Arazketa)
  - I : Info (Informazioa)
  - W : Warning (Abisua)
  - E : Error (Errorea)
  - F : Fatal (Larria)
  - S : Silent (Isila, lehentasun handiena, ezer ez da inprimatzen)

Ondokoa logcat exekutatzean lortzen diren lerroen adibide bat da:

I/ActivityManager(585): Starting activity: Intent { action=android.intent.action...}

Kasu honetan, lehentasun maila "I" da eta etiketa "ActivityManager". Hau da, Info lehentasun mailako mezu bat da ActivityManager-ean duena jatorria.

logcat-en emaitza erabilgarriago egiteko interesatzen zaizkigun lerroak iragaz daitezke iragazki-expresioak erabiliz. Honela, interesatzen zaizkigun etiketa eta lehentasun-maila konbinazioarekin bat datozen mezuak soilik bistaratuko zaizkigu. Iragazkiek etiketa:lehentasuna formatua izaten dute, non etiketa hori duten eta gutxienez zehaztutako lehentasun maila duten mezuak bakarrik bistaratuko diren. Iragazki bat baino gehiago erabil daitezke hutsunez banatuta.

Adibidea:

```
adb logcat ActivityManager:I MyApp:D *:S
```

Kasu honetan, bistaratuko diren mezuek ondoko baldintzetako bat bete beharko dute:

- "ActivityManager" etiketa eta lehentasuna "info" edo altuagoa izatea
- "MyApp" etiketa eta lehentasuna "Debug" edo altuagoa
- *:S, edozein etiketa izan eta gutxieneko lehentasuna "silent" izatea. Honi esker, "View" eta "MyApp" etiketak dituzten mezuak bakarrik agertuko direla ziurtatzen dugu.

Hurrengo adibideak berriz "warning" lehentasun maila edo altuagoa duten mezuak bakarrik bistaratuko ditu:

```
adb logcat *:W
```

4.6.5 Aplikazioen lokalizazioa

4.6.5.1 Oinarri teorikoak

Baliabide lehenetsiak

Testu kate lehenetsiak *res/values/strings.xml* fitxategian daudenak dira, aplikazioa gailuaren *locale*-rako egokitutako baliabiderik aurkitzen ez duenean balio lehenetsiak erabiltzen ditu. Fitxategi lehenetsia falta bada edo aplikazioak behar duen kate bat falta badu errorea emango du eta gelditu egingo da.

Adibidea:

Suposa dezagun aplikazio soil daukagula bi kate bakarrik dituela: text_a eta text_b. Demagun euskarara lokalizatutako baliabide fitxategi bat duela (res/values-eu/strings.xml) bi kateen euskarazko bertsioa definitzen dituena eta baliabide lehenetsientzako beste fitxategi bat duela (res/values/strings.xml), text_a-rentzat definizioa duena baina ez text_b-rentzat.

- Aplikazio hau arazorik gabe konpilatu ahal izango dugu. Eclipse bezalako IDE batek ez digu abisurik emango baliabide bat falta zaigulako.
- Aplikazioa euskarazko *locale* bat duen Android gailu batean abiaraztean ez da arazorik egongo, euskarazko baliabide fitxategiak bi kateen definizioa baitu.
- Baina euskara ez den hizkuntza bateko *locale*-a duen gailu batean errore mezu bat agertuko da Force Close botoiarekin eta ezingo dugu aplikazioa kargatu.

Horrelakorik gerta ez dadin beharrezkoa da balio lehenetsiak dituen fitxategia existitzea (res/values/strings.xml) eta beharrezko kate guztiak definituta egotea bertan. Berdin gertatzen da gainerako baliabide motekin ere (erabiltzaile interfazearen diseinuak, irudiak, animazioak, ...), baliabide lehenetsien fitxategi multzo bat edukitzea beharrezkoa da.

- **res/values/strings.xml** : Testu kate lehenetsiak fitxategi honetan gorde behar dira. Orokorrean ingelesezko kateak.
- **res/drawable/** : Direktorio hau derrigorrezkoa da eta gutxienez fitxategi bat izango du, aplikazioak Market-ean duen ikonoa.
- **res/layout/** : Direktorio hau derrigorrezkoa da eta erabiltzaile interfazearen diseinu lehenetsiak egon behar du bertan.
- **res/anim/** : Direktorio hau derrigorrezkoa da res/anim-<kalifikatzailea> direktoriorik badago.
- **res/xml/** : Direktorio hau derrigorrezkoa da res/xml-<kalifikatzailea> direktoriorik badago.
- **res/raw/** : Direktorio hau derrigorrezkoa da res/raw-<kalifikatzailea> direktoriorik badago.

Nola sortu baliabide alternatiboak

Aplikazio baten lokalizazioaren zatirik garrantzitsuena hizkuntza desberdinetarako testu kate alternatiboak eskaintzean datza. Kasu batzuetan beharrezkoa izango da irudi, soinu, diseinu eta bestelako baliabide alternatiboak sortzea.

Adibidea:

Suposa dezagun gure aplikazioaren hizkuntza lehenetsia ingelesa dela eta testu osoa lokalizatu nahi dugula frantsesera eta guztia izenburua ezik japonierara. Kasu honetan 3 strings.xml fitxategi

beharko ditugu, bakoitza dagokion direktorioan:

1. res/values/strings.xml: Erabiltzen diren kate guztientzat ingelesezko testua duen fitxategia, baita "title" katearentzat ere.
2. res/values-fr/strings.xml: Kate guztientzako frantsesezko testua, baita "title" katearentzat ere.
3. res/values-ja/strings.xml: Kate guztientzako japonierazko testua, "title" katearentzat ezik.

Java kodean R.string.title erabiltzen den lekuan ondoko ordezkapenak emango dira:

- Frantsesezko *locale*-a: res/values-fr/strings.xml fitxategian agertzen den "title" katea erabiliko da (frantsesezko lokalizazioa).
- Beste edozein hizkuntzatako *locale*-a: res/values/strings.xml fitxategian agertzen den "title" katea erabiliko da (lehenetsia).

Japonierazko *locale*-aren kasuan Android-ek res/values-ja/strings.xml fitxategian "title" katea bilatzen saiatuko da lehenik eta ez duenez aurkituko res/values/strings.xml-ko balio lehenetsia kargatuko du.

Zein baliabidek du lehentasuna?

Gailu baten ezarpenekin bat datozen baliabide fitxategi bat baino gehiago badaude, arau jakin batzuk jarraitzen ditu Androidek zein erabili erabakitzeko. Kasu gehienetan *locale*-a nagusitzen da.

Adibidea:

Demagun aplikazio batek grafiko multzo lehenetsi bat duela eta beste bi grafiko multzo ezaugarri desberdinetarako gailuentzat egokituak:

- res/drawable/ Grafiko lehenetsiak
- res/drawable-small-land-stylus/ Stylus bidezko sarrera duen eta QVGA motako dentsitate-baxuko pantaila modu horizontalean duen gailurako egokitutako grafikoak.
- res/drawable-ja/ Japonierarako egokitutako grafikoak.

Gailua japonieraz erabiltzeko konfiguratuta badago res/drawable-ja/ direktorioko grafikoak erabiliko ditu, stylus bidezko sarrera eta QVGA motako dentsitate-baxuko pantaila modu horizontalean dituen gailua izan arren.

Salbuespena: *Locale*-ak baina lehentasun handiagoa duten bakarrak MCC eta MNC dira (*Mobile Country Code* eta *Mobile Network Code*).

Adibidea:

Demagun ondorengo egoera:

- Aplikazioak R.string.text_a deitzen du kodetik.
- Bi baliabide fitxategi daude eskuragarri:
  - res/values-mcc404/strings.xml, non text_a-ri dagokion testua dagoen aplikazioaren hizkuntza lehenetsian, kasu honetan ingelesez.
  - res/values-hi/strings.xml, non text_a-ri dagokion testua dagoen hindueraz.

Gailua hindua erabiltzeko konfiguratuta egon arren res/values-mcc404/strings.xml-tik ingelesezko testua kargatuko da text_a testuarentzat. Androidentzat lehentasun gehiago du MCC-a bat etortzeak

hizkuntza bat etortzeak baino.

Javako kodetik baliabideak deitzea

R.baliabide_mota.baliabide_izena edo android.R.baliabide_mota.baliabide_izena erabiliz lor daitezke baliabideen edukiak.

Lokalizazio estrategiak

- Edozein *locale*-rekin lan egiteko gauza izatea kontutan izan aplikazioa diseinatzerakoan.
- Garatzen dugun aplikazioa erabiliko duten gailuen ezaugarriak aurreikustea ezinezkoa da.
- Garrantzitsua da baliabide lehenetsiak izatea erabiltzen diren baliabide guztientzat, hau da, ziurtatu res/drawable/ eta res/values/ direktorioek beharrezko testu eta grafiko guztiak dituztela.

4.6.5.2 Aplikazioen lokalizazioan erabilitako tresnak

4.6.5.2.1 Euskalbar

Euskalbarrek hitzak itzultzen laguntzen du euskaratik ingelesera, frantsesera, gaztelaniara eta japonierara, eta ingelesetik, frantsesetik, gaztelaniatik eta latinetik euskarara. Firefoxen azaltzen den barra berri bati esker, bilaketak modu errazean egiten dira euskarazko hainbat hiztegi eta baliabidetan (Euskalterm, Elhuyar, Hiztegia3000, Euskaltzaindia, ItzuL posta zerrendaren artxiboa, Harluxet, Mokoroa, ZT Corpusa, Opentrad itzultzaile automatikoa eta XUXEN web zuzentzaile ortografikoa). Barra moduan ez ezik, aukeratutako testu baten gainean egin daitezke bilaketak, eskuin-klika erabiliz horretarako. Bilaketa arruntez gain, bilaketa konbinatuak ere egiten ditu Euskalbarrek, hainbat hiztegien arteko bilaketak orrialde bakarrean erakutsiz.

GNU General Public License, 3.0 bertsioa erabiliz argitaratua.

Itzulpenak egiten ari naizenean agertutako zalantzak argitzeko erabiliko dut.

Webgunea: http://euskalbar.eu

Kode biltegia: http://code.google.com/p/euskalbar/

4.6.5.2.2 Launchpad

Launchpad softwarea, bereziki librea, garatu eta mantentzea ahalbideratzen duen webgune eta web aplikazioa da, Canonical Ltd.-ek garatu eta mantendutakoa.

2011ko uztailean 23500 proiektu aterpetzen zituen.

Kritika dezente jaso zituen bere garaian software librea ez izateagatik, azkenean, iturburu kodea 2009ko uztailean argitaratu zen GNU Affero GPL lizentziapean.

Androiderako ConnectBot aplikazioaren itzulpena egiteko erabiliko dut.

Webgunea: https://launchpad.net/

Kode biltegia: https://code.launchpad.net/launchpad

4.6.6 Android Market

4.6.6.1 Mugak eta arazoak

- Android Market aplikazioa ez da software librea.
- Google-ren bateragarritasun betebeharrak betetzen dituzten Android gailuak bakarrik daude baimenduta Android Market aplikazioa instalatu eta erabiltzeko.
- Smartphone-ak ez diren gailuei ezarritako mugak.
- Galarazitako aplikazioak:
  - Tethering aplikazioak Market-etik kanporatu zituen Googlek 2009ko Martxoaren 31an. Beranduago berriz onartu zituen, T-Mobile USA sareko erabiltzaileentzat ezik.
  - 2011ko apirilaren 5ean Grooveshark aplikazioa kanporatua izan zen Marketetik zehaztu gabeko lizentzia urraketak zirela medio. Hala ere, Grooveshark-en web orritik deskargatu eta instala daiteke aplikazio hau arazorik gabe.
  - 2011ko lehen hiruhilekoan Easy Tether eta root baimen beharrik gabeko beste tethering aplikazioak blokeatzen hasi zen Verizon Wireless Android gailuetan.
  - 2011ko maiatzaren 29an zenbait bideojoko emulatzaileren garatzaileen kontuak blokeatzen hasi zen Google, arrazoirik jakinarazi gabe.

4.6.6.2 Lizentzia libreko software biltegia: Fdroid

FDroid Software Biltegia Android plataformarako lizentzia libreko softwarearen biltegi bat da. Aplikazio bakoitzaren hainbat bertsiori buruzko informazioa dago zerbitzarian eta Android bezeroak aplikazioa aukeratu, instalatu eta eguneratuta mantentzea errazten du.

Web nabigatzaile baten bidez ere araka daiteke software biltegia eta baita aplikazioak deskargatu ere: http://f-droid.org/repository/browse/

Bertako aplikazioak benetan libreak direla ziurtatzeko beren iturburu kodeak aztertzen dira eta publikatutako iturburutik abiatuta eraikitzen dira apk berdina lortzen dela egiaztatzeko.

Baimenik gabe erabiltzailearen aktibitateei buruzko datuak (adibidez, Google Analytics bidez) edo erabiltzailearen portaerei buruzko (adibidez, publizitate plataformak) txostenak bidaltzen dituzten aplikazioak guztiz galarazita daude FDroid software biltegian, libreak ez diren sareko zerbitzuekin elkarrekintza baitute lehen helburu.

Aplikazio berri bat gehitu nahi izanez gero ondoko eskaera-orria bete behar da:

http://f-droid.org/repository/submit

Ondoko loturatik deskargatu behar da (ez daukate asmorik Android Market-ean eskaintzeko) eta Marketekoak ez diren aplikazioak instalatzeko aukera gaitu Android gailuan:

http://f-droid.org/FDroid.apk

F-Droid GPLv2 lizentziadun software librea da.

Webgunea: http://f-droid.org

Kode biltegia: http://gitorious.org/f-droid

4.6.6.3 Beste aukerak:

4.6.6.3.1 http://slideme.org

2008ko apiriletik daude eskuragarri web orria eta Androiderako bezeroa. Aplikazioak publikatzea doan da.

Aplikazio biltegia: http://slideme.org/applications

SlideMe Application Manager aplikazioa:

http://slideme.org/application/sam

http://slideme.org/sam.apk

Jabedun lizentzia du.

4.6.7 Android ez da smartphone eta tabletetarako soilik

4.6.7.1 Google TV

Google TV Googleren Smart TV/Connected TV plataforma da. 2010eko maiatzaren 20ean iragarri zuten Google I/O ekitaldian eta Google, Intel, Sony eta Logitech-ek garatu zuten elkarlanean. Google TV-k Android sistema eragilea eta Google Chrome nabigatzailearen GNU/Linux-erako bertsioa integratzen ditu. Ofizialki 2010eko urriaren 6an kaleratu zen, Sony eta Logitech-en gailuetan.

Web ofiziala: http://www.google.com/tv/

4.6.7.1.1 Kritikak

Google TV kritika txar ugari jaso ditu. Kritiketako batzuk ondokoak dira:

- Behar bezala bukatu gauzatu gabeko produktu bat edo beta bat dirudiela komentatu izan da.
- Erabiltzaile interfazea kontraesankorra eta nahasgarria da.
- Kodekak falta ditu.
- Teknofiloentzako interesgarria izan daiteke baina ez erabiltzaile arruntentzat.
- Konplexuegia da.

2011ko uztailean www.techcrunch.com webeko Matt Burns-ek aipatzen zuen bezala Google TV gainbehera doala dirudi.

Logitech-ek bere Google TV aparailuak saldu ezinik dabilela dirudi. Salneurria 250$etatik 99$era jaitsi du, streaming bidezko bideo konpainia handiek Google TV bidezko konexioak blokeatzen hasi ondoren. Gainera 2011ko lehen hiruhilabetekoan produktuen itzulerek salmentak gainditu dituzte.

2011ko abuztuan Sony-k ere Google TVdun bere HDTV telebisten salneurria jaitsi behar izan du.

4.6.7.2 Scandinavia telebista

Suediako People of Lava enpresak Scandinavia izeneko Android sistema eragiledun munduko lehen telebista merkaturatu zuen.

Ezaugarriak:

- Android 1.5 Cupcake erabiltzen du eta hainbat aplikazio dakartza: interneteko nabigatzailea, You Tubeko bideoak ikusteko aplikazioa, Google Maps, ...
- Gama altuko telebistak dira, eskuz muntatutakoak eta 42tik 55 hazbeterainoko pantailadunak.
- Garestiak: Salneurriak 2500 eta 4000 euro artean daude.

Webgunea: http://www.peopleoflava.com/television/scandinavia/

4.6.7.2.1 Scandinaviarentzako aplikazioen garapena

Androiderako aplikazioen garatzaileak plataforma berri honetarako lan egitera bultzatu nahi dituzte.

http://developer.peopleoflava.com/ web orrian garapenerako gidalerroak, aplikazioak garatzen hasteko tutoriala, adibideak, zalantzak argitzeko foroa, FAQ eta abar daude.

Tutorialean azaltzen denez, telebista hauen bereizmena 1080×720 da baina Android emulatzaileak ez du onartzen horrenbesteko bereizmenik. Hori dela eta, <android-sdk>/platform/android-3/images/ katalogoan aurkitzen den Android 1.5 (API3) emulatzailearen irudia ordezkatu beharra dago beraiek eskaintzen dutenarekin:

http://developer.peopleoflava.com/wp-content/uploads/2011/02/kernel-qemu-1.zip

Garapenean Eclipse erabili nahi bada pilaren tamaina handitzea gomendatzen da tutorialean.

4.6.7.3 ANDI-ONE urruneko agintegailu unibertsala

Koreako Conspin etxeak merkaturatutako ANDI-ONE da Android sistema eragilea duen munduko lehen urruneko agintegailu unibertsala.

Ezaugarriak:

- Android 2.1 erabiltzen du
- 3,5 hazbeteko ukipen pantaila
- 50 gailu kontrola ditzake infragorri (IR), irrati frekuentzia(RF) eta WiFi bidez
- Interneterako konexioa
- VoIP telefono bezala erabiltzeko aukera
- 2 GBeko flash memoria eta 32 GBerainoko SD txartela erabiltzeko aukera
- Garestia: Gomendatutako salneurria 350$ da

Webgunea: http://www.conspin.com/conspin/andione.html

4.6.7.4 I'mWatch erlojua

Italiako Blue Sky enpresak merkatuko du munduko Android sistema eragiledun lehen smartwatch-a.

Ezaugarriak:

- Android sistema eragilearen bertsio berezia (1.6aren antzekoa)
- Erloju digital edo analogiko bistaratzeak
- 1,54 hazbeteko pantaila

- Telefonoarekin elkarrekintza bluetooth bidez: Deiak, SMSak eta e-postak jasotzean abisua eta aurrebista (nor ari den deitzen, mezua bidaltzen)
- 4 GBeko flash memoria
- Garestia: 249€

Webgunea: http://www.imwatch.it/

4.6.7.5 Nook Simple Touch e-liburu irakurgailua

Nook enpresaren bigarren belaunaldiko Android sistema eragilea erabiltzen duen e-liburu irakurgailua.

Ezugarriak:

- Android 2.1 sistema eragilea
- 6 hazbeteko e-tintazko ukipen pantaila
- 2 GB flash memoria, 32 GBerainoko SD txartela erabiltzeko aukera
- Wi-Fi
- Root baimena eskuratuz gero (bermea galtzen da) Android aplikazioak instala daitezke
- Salneurria: 139$

Webgunea: http://www.barnesandnoble.com/nook/index.asp?PID=35699

4.6.8 Bibliografia

Android Sistema Eragilea

- http://en.wikipedia.org/wiki/Android_%28operating_system%29
- http://www.gubatron.com/blog/2010/05/23/how-many-lines-of-code-does-it-take-to-create-the-android-os/
- http://developer.android.com/guide/basics/what-is-android.html

Android %100 librea: Replicant proiektua

- http://replicant.us/about/
- http://replicant.us/faq/
- Replicant-en wikia eta trac:

  http://trac.osuosl.org/trac/replicant
- HTC Dream gailuen jabedun softwarearen zerrenda osoa:

  http://trac.osuosl.org/trac/replicant/wiki/HTCDreamProprietaryDrivers

Androiderako aplikazioen garapenerako tresnak

- **Android SDK (Software Depelopment Kit)**:
  - http://en.wikipedia.org/wiki/Android_software_development#Android_SDK
  - http://developer.android.com/sdk/index.html
  - http://developer.android.com/guide/developing/devices/index.html

    - http://developer.android.com/guide/developing/tools/adt.html
    - http://developer.android.com/guide/developing/devices/emulator.html
    - http://developer.android.com/guide/developing/tools/emulator.html
- **Android NDK (Native Development Kit)**:
    - http://en.wikipedia.org/wiki/Android_SDK#Native_Development_Kit
    - http://developer.android.com/sdk/ndk/index.html
    - http://developer.android.com/sdk/ndk/overview.html
    - NDK erabiltzen duten proiektuen garapena Eclipse erabiliz: http://codemaemo.appforce.org/2010/07/tutorial-using-eclipse-for-ndk-projects/
    - Nola erabili NDK Androideko aplikazio batean C-ko kodea erabiltzeko: http://marakana.com/forums/android/examples/49.html
- **Eclipse ADT plugina**:
    - http://developer.android.com/sdk/eclipse-adt.html
- **Android Virtual Devices (AVD)**:
    - http://developer.android.com/guide/developing/devices/index.html
    - http://developer.android.com/guide/developing/devices/managing-avds.html
    - http://developer.android.com/guide/developing/tools/emulator.html
- **Android Debug Bridge (ADB)**:
    - http://developer.android.com/guide/developing/tools/adb.html

Androiderako aplikazioen lokalizazioa

- http://developer.android.com/guide/topics/resources/localization.html
- **Hello, L10N tutoriala**: http://developer.android.com/resources/tutorials/localization/index.html
- **Launchpad**:
  http://en.wikipedia.org/wiki/Launchpad_%28website%29
  http://blog.launchpad.net/general/launchpad-is-now-open-source

Android Market

- http://en.wikipedia.org/wiki/Android_market
- **Mugak eta arazoak**:
    - http://www.wired.com/gadgetlab/2010/03/android-devices-crave-googles-attention/
    - http://gigaom.com/2010/04/04/google-is-missing-an-android-opportunity-on-non-smartphones/
    - http://www.techradar.com/news/phone-and-communications/mobile-phones/google-android-not-optimized-for-tablets--715550

- **Lizentzia libreko software biltegia: F-droid**
  - http://f-droid.org/about/
  - http://en.wikipedia.org/wiki/List_of_free_software_Android_applications#Free_software_repositories
- **Beste aukerak: SlideMe**
  - http://slideme.org/about-slideme
  - http://slideme.org/faq

Android ez da smartphone eta tabletetarako soilik

- **GoogleTV**
  - http://en.wikipedia.org/wiki/Google_TV
  - Iragarpen ofiziala Googleren blogean:

    http://googleblog.blogspot.com/2010/05/announcing-google-tv-tv-meets-web-web.html
  - http://www.engadget.com/2010/10/29/google-tv-review/
  - Comedy Central eta MTV Google TV blokeatzen:

    http://news.cnet.com/8301-17938_105-20023547-1.html
  - Kritikak
    - http://www.nytimes.com/2010/11/18/technology/personaltech/18pogue.html?pagewanted=all
    - http://techcrunch.com/2011/07/28/logitech-looses-big-on-google-tv-revue-price-cut-from-250-to-99/
    - http://bits.blogs.nytimes.com/2010/10/21/big-networks-block-web-shows-from-google-tv/
- **Scandinavia telebista**
  - http://www.peopleoflava.com/television/scandinavia/
- **ANDI-ONE urruneko agintegailu unibertsala**
  - http://www.conspin.com/conspin/andione_feature.html
  - http://www.engadget.com/2011/04/12/andi-one-universal-remote-runs-android-2-1-does-more-than-chang/
  - http://www.talkandroid.com/36626-conspin-introduces-the-andi-one-the-worlds-first-android-universal-remote-control/
- **I'mWatch**
  - http://www.imwatch.it/en/smartwatch/features/
- **Nook Simple Touch**
  - http://en.wikipedia.org/wiki/Nook_Simple_Touch

4.7 Aztertutako tekniketarako Androiderako aplikazio libreak

4.7.1 Port Knocking / Single Packet Authorization

4.7.1.1 Android-Fwknop

Fwknop *single packet authorization* bezeroaren Androiderako egokitzapena da, Max Kastanas-ek egina, Damien Stuart-en libfko erabiliz.

SPA paketea bidali eta urruneko makinako fwknop daemonak baliozkotasuna egiaztatu ondoren, Androideko fwknop aplikazioak Connectbot abiarazten du automatikoki irekitako atakan zehar, urruneko makinara SSH bidez konektatzeko.

Android 2.0 edo berriagoa behar du.

GPL 2.0 lizentziapean kaleratua.

Android Market: https://market.android.com/details?id=com.max2idea.android.fwknop

Kode biltegia: http://trac.cipherdyne.org/trac/fwknop/browser/trunk/android

4.7.2 Wake On Lan (WOL)

4.7.2.1 Wake-On-Lan

Androiderako Wake On LAN aplikazio soil bat da. Bidalitako paketeen historia gordetzen du MAC helbideak behin eta berriz idatzi beharrik ez izateko.

Android 1.5 edo berriagoa behar du.

BSD lizentziapean kaleratua.

Android Market: https://market.android.com/details?id=net.mafro.android.wakeonlan

Kode biltegia: https://github.com/mafrosis/Wake-On-Lan

4.7.3 SSH

4.7.3.1 ConnectBot

ConnectBot Androiderako Secure Shell (SSH) bezero libre ahaltsua da, Kenny Root eta Jeffrey Sharkey-k garatua. Aldibereko SSH saioekin lan egin dezake, tunel seguruak sortu eta beste aplikazioen artean kopiatu/itsatsi.

Android 1.5 edo berriagoa behar du.

Apache 2.0 lizentziapean kaleratua.

Android Market: https://market.android.com/details?id=org.connectbot

Kode biltegia: http://code.google.com/p/connectbot/

4.7.3.2 SSH Tunnel

SSH Tunnel ConnectBot eta Dropbear/OpenSSH(Beta Branch) aplikazioetan oinarritutako Androiderako SSH tunel aplikazio bat da.

Android SSH Tunnel aplikazioak modu erraz batean SSH tunel bat sortzeko aukera eskaintzen du,

sistema osoarentzat edo aplikazio batentzat. Helburu ofiziala Txinako erabiltzaileei "Suhesi Handia" gainditzeko aukera eskaintzea den arren edozeinek erabil dezake modu pribatuan nabigatzeko.

SSH tunela bi makina Secure Shell (SSH) bidez konektatzea da, bien artean datu transmisio zifratua ahalbideratzeko. Datuak berak ez dira zifratzen baina zifratutako tunel batean zehar doazenez kanpotik entzun nahian ari denarentzat ulertezinak dira. Honi esker posible da suhesi mugatzaileetan zehar komunikazioak ezartzea, suhesiak ez baitu jakingo zer bidaltzen ari garen.

Hala ere, erabilera bakarra ez da pribatutasuna ziurtatzea. Datuak bidaltzeko SSH tunel bat erabiltzean, datuen jatorria tunelaren bukaerako zerbitzarian dagoela dirudi. Honi esker, AEBetara mugatutako zerbitzu bat erabil daiteke Europatik, adibidez Pandora, horretarako AEBetako zerbitzari batetara tunel bat sortuz.

Android 1.6 edo berriagoa behar du.

GPLv3 lizentziapean kaleratua.

Android Market: https://market.android.com/details?id=org.sshtunnel

Kode biltegia: http://code.google.com/p/sshtunnel/

4.7.4 VNC

4.7.4.1 Androiderako VNC bezeroak

4.7.4.1.1 Android-vnc-viewer

Android gailutik konputagailuaren mahaigaina ikusi eta kontrolatzea ahalbideratzen duen aplikazioa da. VNC zerbitzari gehienetara konekta daiteke, hala nola, TightVNC, RealVNC, x11vnc eta OS/X-eko *Apple Remote Desktop*.

Android 1.5 edo berriagoa behar du.

GPLv2 lizentziapean kaleratua.

Android Market: https://market.android.com/details?id=android.androidVNC

Kode biltegia: http://code.google.com/p/android-vnc-viewer/

4.7.4.2 Androiderako VNC zerbitzariak

4.7.4.2.1 droid-vnc-server

Androiderako VNC zerbitzari honek gure gailua USB edo Wifi bidez kontrolatzeko ordenagailua erabiltzea ahalbideratzen du. Oraindik beta egoeran dago.

Ezaugarriak:

- Pasahitz bidezko autentifikazioa
- Errotatu/Eskalatu
- Wifi, USB eta 3G
- Sagua eta teklatuaren emulazioa
- Arbela erabil daiteke

ADI: Android gailuan root baimenak behar ditugu erabili ahal izateko.

Android 1.5 edo berriagoa behar du.

Software librea da (ez dakit zein lizentzia erabiltzen duen).

Android Market: https://market.android.com/details?id=org.onaips.vnc

Kode biltegia: https://github.com/oNaiPs/droid-VNC-server

4.7.4.2.2 android-vnc-server

Konpilatutako binarioa bai baina ez dut apk-rik aurkitu:

http://code.google.com/p/android-vnc-server/downloads/list

Android Market-ean ere ez dut aurkitu. Hala ere, proiektua aktibo dagoela dirudi:

http://code.google.com/p/android-vnc-server/updates/list

android-vnc-n oinarritua dago. android-vnc-ren kode biltegia ondokoa da:

http://code.google.com/p/android-vnc/

Azken proiektu hau utzita dagoela dirudi, azken aldaketak 2007ko abenduan egin baitziren. Hemen ere ez dut apk-rik aurkitu.

GPLv2 lizentziapean kaleratua.

Kode biltegia: http://code.google.com/p/android-vnc-server/

4.7.5 Androiderako aplikazio gehigarriak

4.7.5.1 BusyBox

BusyBox "Kapsulatutako Linux-erako Suitzar Labana" deitua izan da.

UNIX lanabes arrunten bertsio txikiak biltzen dituen aplikazioa da BusyBox. GNU fileutils, shellutils, eta abarretan aurkitzen diren tresnen ordezkoak eskaintzen ditu. GNUko lehengusuak baina aukera gutxiago eskaintzen dituzte orokorrean, baina antzeko funtzionalitatearekin. Kapsulatutako gailuetarako pentsatuta daudenez baliabide murritzetarako optimizatuta daude. Gainera guztiz modularrak dira eta komandoak edo ezaugarriak kendu edo gehi daitezke konpilatzean, kasu bakoitzeko beharretara egokitzea errazteko.

Android 1.0 edo berriagoa behar du.

GPLv2 lizentziapean banatzen da.

Android Market: https://market.android.com/details?id=stericson.busybox

4.7.5.2 Android Terminal Emulator

Aplikazio hau Digital Equipment Corporation VT-100 terminalaren emulatzaile bat da, GNU/Linux-eko komando lerroko programak egikaritzeko aukera eskaintzen du. Adibidez, vi eta Emacs behar bezala bistaratzeko gauza da.

Android Shell komandoen erreferentzia: https://github.com/jackpal/Android-Terminal-Emulator/wiki/Android-Shell-Command-Reference

Android 1.5 edo berriagoa behar du.

Apache 2.0 lizentziapean banatzen da.

Android Market: https://market.android.com/details?id=jackpal.androidterm

Kode biltegia: https://github.com/jackpal/Android-Terminal-Emulator

4.7.5.3 android-screencast

Android gailu bat gure ordenagailutik kontrolatu, bere pantaila leiho batean ikusi eta sagua/teklatua erabili ahal izatea ahalbideratzen duen mahaigaineko aplikazio bat da. Plataforma anitza da (Windows/Linux/MacOS) eta android gailu guztientzat balio du.

Ezaugarriak:

- Sagua eta teklatua erabiltzeko aukera soilik Root baimenak baditugu gailuan
- Ikuspegi horizontala (eskuineko botoian klik eginez)
- Bideoan grabatzeko aukera
- Oinarrizko fitxategi-nabigazioa
- Apache 2.0 lizentzia

Mugak:

- Mantsoa (4-5 fps)
- Tekla guztiak ez daude mapeatuta

Apache 2.0 lizentziapean banatzen da.

Kode biltegia: http://code.google.com/p/androidscreencast/

4.7.6 Bibliografia

Fwknop

- http://cipherdyne.org/blog/2011/01/single-packet-authorization-on-android.html
- Garapenerako oharrak, beharrezko softwarea, nola prestatu garapen ingurunea eta abar azaltzen dira ondoko web orrian: http://trac.cipherdyne.org/trac/fwknop/browser/trunk/android/README

SSH Tunnel

- http://lifehacker.com/5799888/ssh-tunnel-is-the-easiest-way-to-tunnel-on-your-android-device

droid-vnc-server

- Erabilera gida: http://opensourceexcedio.wordpress.com/2010/10/28/droid-vnc-server/
- Garatzailearen bloga: http://www.onaips.com/wordpress/?cat=83

Busybox

- http://www.busybox.net/

Android Terminal Emulator

- https://github.com/jackpal/Android-Terminal-Emulator/wiki

4.8 Ikerketa lanean erabilitako beste aplikazioak

4.8.1 Eclipse

Eclipse software garapenerako ingurune lengoaia-anitza da, garapen ingurune integratu (IDE) eta plugin sistema batez osatua. Erabilera lehenetsia Java lengoaian programatzea bada ere, hainbat plugin erabiliz beste programazio lengoaietan programatzeko ere erabil daiteke, hala nola, Ada, C, C++, COBOL, Perl, PHP, Python, R, Ruby, Scala, Clojure, Groovy eta Scheme.

Eclipse Public License lizentziapean banatzen da.

Webgunea: http://eclipse.org/

4.8.2 LibreOffice Writer

LibreOffice Writer LibreOffice software bildumako testu prozesatzailea da. LibreOffice *Document Foundation*-ek garatzen duen software libreko bulego aplikazioen bilduma da, OpenOffice.org-en *fork* bezala sortua.

"*The Document Foundation*" taldea 2010eko irailaren 28an OpenOffice.org-eko hainbat kidek sortu zuten Oracle-k OpenOffice.org baztertu edo mugatuko zuenaren beldur.

GNU Lesser General Public License (LGPL) lizentziapean kaleratutako software librea da.

Webgunea: http://www.libreoffice.org/

4.8.3 VirtualBox

x86 makinen birtualizaziorako sofware pakete bat da.

VirtualBox-eko Ubuntu 10.04 makina birtual bat erabili dut lehen probak egiteko. Makina birtuala zerbitzari bezala erabili dut eta nire makina bezero bezala.

Webgunea: http://www.virtualbox.org/

4.8.4 Gedit

Gedit GNOME mahaigain ingurunearen helburu orokorreko testu editorea da.

XML fitxategien itzulpenak egiteko eta oharrak hartzeko erabiliko dut.

GPL lizentzipean kaleratuko software librea da.

http://projects.gnome.org/gedit/

4.8.5 Bibliografia

Eclipse

- http://en.wikipedia.org/wiki/Eclipse_%28software%29

LibreOffice Writer

- http://en.wikipedia.org/wiki/LibreOffice_Writer

Virtualbox

- http://en.wikipedia.org/wiki/VirtualBox

Gedit

- http://en.wikipedia.org/wiki/Gedit

5 Garapen teknikoa eta probak

5.1 Port Knocking-a knockd-rekin

Zerbitzarian knockd eta sshd zerbitzariak behar dira. sshd dagoeneko instalatuta dagoela suposatuko dugu, SSH-ri dagokion atalean aztertuko dugu nola instalatu eta konfiguratu.

ifconfig erabiliz zerbitzariaren IP-a zein den lortu behar dugu gero bezerotik konektatzeko, 192.168.56.101 da nire kasuan.

SSH zerbitzaria martxan jarriko dugu:

```
sudo /etc/init.d/ssh start
```

Eta sshd daemona entzuten dagoela egiaztatu:

```
sudo netstat -anlp | grep sshd
```

tcp 0 0 0.0.0.0:22 0.0.0.0: LISTEN 1834/sshd*

tcp6 0 0 :::22 ::: LISTEN 1834/sshd*

nmap erabiliz zerbitzariko 22 ataka entzuten dagoen egiaztatuko dugu bezerotik:

```
nmap -p 22 192.168.56.101
```

Starting Nmap 5.00 (http://nmap.org) at 2011-08-17 19:45 CEST

Interesting ports on 192.168.56.101:

PORT STATE SERVICE

22/tcp open ssh

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds

Ondoren, bezerotik zerbitzarira SSH bidez konektatu gaitezkeela egiaztatuko dugu:

```
ssh asier@192.168.56.101
```

Jarraian, suhesia prestatuko dugu zerbitzarian. Lehenik eta behin orain arte erabilitako suhesiaren kopia bat egingo dugu:

```
sudo iptables-save -c > /etc/iptables-save
```

Berreskuratu nahi izanez gero:

```
cat /etc/iptables-save | sudo iptables-restore -c
```

Zerbitzariko suhesiaren arau guztiak ezabatuko ditugu:

```
sudo /sbin/iptables -F
```

```
sudo /sbin/iptables -F -t nat
sudo /sbin/iptables -X
```

Eta ondoren, zerbitzarian suhesiaren arauak ezarri:

```
sudo /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED
-j ACCEPT
sudo /sbin/iptables -A INPUT  -j DROP
```

Berriz ere, nmap-ekin 22 ataka orain itxita dagoela egiaztatu bezerotik:

```
nmap -p 22 192.168.56.101
```

Starting Nmap 5.00 (http://nmap.org) at 2011-08-17 19:49 CEST

Note: Host seems down. If it is really up, but blocking our ping probes, try -PN

Nmap done: 1 IP address (0 hosts up) scanned in 3.08 seconds

Eta SSH bidez konektatzen saiatuko gara baina orain ezingo dugu ezarri dugun suhesiak galaraziko baitigu:

```
ssh asier@192.168.56.101
```

ssh: connect to host 192.168.56.101 port 22: Connection timed out

Hurrengo pausoa, knockd daemona konfiguratzea izango da, etc katalogoko knockd.conf fitxategiaren bidez. Hau da */etc/knockd.conf* fitxategiaren edukia:

[options]

UseSyslog

[openSSH]

sequence = 7000,8000,9000

seq_timeout = 5

command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

tcpflags = syn

[closeSSH]

sequence = 9000,8000,7000

seq_timeout = 5

command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

tcpflags = syn

Oharra: Jarraitu dudan tutorialean -I-ren ordez -A agertzen da bezeroari 22 ataka irekitzeko iptables-eko arauan, hori dela eta, araua bukaeran gehitzen da eta bezeroak ezin du konektatu, beti DROP araua aplikatzen baita, arau berria baino lehenago dagoelako. -I-rekin berriz arau berria arau-zerrendan lehenengo agertzen da eta bezeroa konekta daiteke.

7000, 8000, 9000 sekuentziak 22 ataka zabalduko du bezeroaren IParentzat eta 9000, 8000, 7000

sekuentziak berriz itxi.

Knockd-ren fitxategi lehenetsiaren edukia berriz ondokoa da:

```
##################################################
#
# knockd's default file, for generic sys config
#
##################################################
# control if we start knockd at init or not
# 1 = start
# anything else = don't start
#
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1
# command line options
KNOCKD_OPTS="-i eth1"
```

Ondoren, knockd daemona abiaraziko dugu:

```
sudo /etc/init.d/knockd start
 * Starting Port-knock daemon knockd                    [ OK ]
```

Bezeroan knock aplikazioa erabiliko dugu ate jotzea gauzatzeko:

```
knock -v 192.168.56.101 7000 8000 9000
```

Ate jotzeari erantzunez suhesiaren arauak aldatu dira, SSH bidezko sarbidea ahalbideratuz bezeroari:

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  192.168.56.1        anywhere            tcp dpt:ssh
ACCEPT     all  --  anywhere            anywhere            state RELATED,ESTABLISHED
DROP       all  --  anywhere            anywhere
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Honi esker, SSH bidez konektatu ahalko gara:

```
ssh asier@192.168.56.101
```

5.1.1 Bibliografia

- https://help.ubuntu.com/community/PortKnocking

5.2 Single Packet Authorization fwknop-ekin

SSH zerbitzari bat beharko dugu, dagoeneko instalatuta daukat eta SSH-ri dagokion zatian azalduko dut nola instalatu eta konfiguratu.

Zerbitzari bezala lan egingo duen ordenagailuan fwknop-server paketea instalatuko dugu:

```
sudo apt-get install fwknop-server
```

Ondoko mezua agertuko zaigu:

"FireWall KNock OPerator" deabrua ez da konfiguratu oraindik. Instalazio prozesu honek fwknopd partekatuta Rijndael gako soil bat erabiltzeko konfigura dezake, baina GnuGP konfigurazio batetara aldatzea gomendatzen da. SPA komunikazioetarako GnuPG konfiguratzeko fwknop dokumentazioan azaltzen diren eskuzko urrats batzuek egin behar dira. Bitartean SPA enkriptazio eta desenkriptaziorako Rijndael erabiltzeak segurtasun dezente bat ematen du.

Konfiguratu fwknop SSH ataka babesteko?

<Bai> *<Ez>*

Bai sakatuko dugu eta ondoren beste mezu bat agertuko zaigu usnatuko den interfazeari buruz galdezka:

Mesedez zehaztu zein Ethernet interfaze ipini behar den modu promiskuoan.

*Usnatuko den interfazea:*_____

<Ados>

Nire kasuan usnatuko den interfazea eth1 da, berau idatzi eta Ados sakatu ondoren, Rijndael bloke zifraketarako gako bat eskatuko digu:

Lehenespen bezala SPA paketeak enkriptazio gako bat behar duen etherneth Rijndael bloke zifraketarekin enkriptatzen dira. Pasahitz honek behintzat 8 karaktereko luzera eduki behar du.

Erabiliko den enkriptazio gakoa:

_____

<Ados>

Gakoa idatzi eta Ados sakatu dut. Instalazioak amaiera arte jarraitu du beste galderarik egin gabe.

man fwknopd egikaritu dut informazio gehiago lortzeko eta bertan irakurri dudanez bi konfigurazio fitxategi daude:

- **/etc/fwknop/fwknop.conf**: Konfigurazio fitxategi nagusia.
- **/etc/fwknop/access.conf**: Ate jotze sekuentziak eta sarbide kontrol direktibak definitzen ditu.

Daemona martxan jartzeko:

```
sudo /etc/init.d/fwknop-server start
```

Gelditzeko:

```
sudo /etc/init.d/fwknop-server stop
```

Berrabiarazteko:

```
sudo /etc/init.d/fwknop-server restart
```

Daemonaren egoera jakiteko:

```
sudo /etc/init.d/fwknop-server status
```

Zerbitzarian portu guztiak itxiko ditugu, portu sekuentzia bidezko port knocking-ean egin bezala:

```
sudo iptables -F
sudo iptables -F -t nat
sudo iptables -X
```

Ondoren, gure arauak ezarriko ditugu, dagoeneko abiarazita dauden konexioak onartuko ditugu, gainerako guztiak galarazi:

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A INPUT -j DROP
```

Ikus ditzagun ezarri ditugun arauak:

```
sudo iptables -L
```

Chain INPUT (policy ACCEPT)

target prot opt source destination

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

DROP all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

Bezero bezala lan egingo duen makinan fwknop-client paketea instalatuko dugu:

```
sudo apt-get install fwknop-client
```

Zerbitzarian fwknop daemona martxan dagoela egiaztatu ondoren, bezeroan ondokoa egikarituko dugu:

```
fwknop -A tcp/22 -s -k 192.168.56.2
```

[+] Starting fwknop client (SPA mode)...

[+] Enter an encryption key. This key must match a key in the file

/etc/fwknop/access.conf on the remote system.

Encryption Key:

[+] Building encrypted Single Packet Authorization (SPA) message...

[+] Packet fields:

Random data: 8152907130579514

Username: asier

Timestamp: 1314653055

Version: 1.9.12

Type: 1 (access mode)

Access: 0.0.0.0,tcp/22

SHA256 digest: HEmzK11F2khMq4B0akY/+J6MpGZPOG7Vhi2hetDUutc

[+] Sending 161 byte message to 192.168.56.2 over udp/62201...

Eskatzen digun encryption key pasahitza lehen zerbitzaria instalatzerakoan zehaztutakoa da, /etc/fwknop/access.conf fitxategian KEY bezala agertzen dena.

Ondoren, /etc/fwknop/access.conf fitxategian FW_ACCESS_TIMEOUT bezala zehaztutako denbora, segundutan, izango dugu SSH bidez zerbitzarira konektatzeko. Balio lehenetsia 30 segundu da.

```
ssh asier@192.168.56.2
```

Suhesiaren arauak nola aldatu diren ikusiko dugu:

```
sudo iptables -L
```

Chain INPUT (policy ACCEPT)

target prot opt source destination

FWKNOP_INPUT all -- anywhere anywhere

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

DROP all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

Chain FWKNOP_INPUT (1 references)

target prot opt source destination

ACCEPT tcp -- 192.168.56.1 anywhere tcp dpt:ssh

GnuPGP autentifikazioa

Segurtasun maila hobetzeko interesgarria da GnuPGP erabiltzea. Bezeroak bere gako pribatua erabili behar luke SPA paketea digitalki sinatzeko eta zerbitzariaren gako publikoa erabili zifratzeko. Ondoren, zerbitzariak bezeroaren gako publikoa eta sinadura digitala erabiliko lituzke paketearen jatorriaren fidagarritasuna ziurtatzeko. Hau da, bai bezeroak bai zerbitzariak bata bestearen gako publikoak ezagutu behar dituzte.

Bezeroan GnuPG gako parea sortu eta gako publikoa fitxategi batetara esportatu behar dugu:

```
gpg --gen-key
```

Ondoren ze gako mota sortu nahi dugun galdetuko digu, balio lehenetsia, RSA, hautatutako dugu:

Please select what kind of key you want:

(1) RSA and RSA (default)

(2) DSA and Elgamal

(3) DSA (sign only)

(4) RSA (sign only)

Your selection? 1

Hurrengo galderan gakoen tamaina zehaztu beharko dugu (1024 eta 2048 artean), balio lehenetsia, 2048 bit, hautatuko dugu:

What keysize do you want? (2048)

Ondoren, gakoa noiz baliogabetuko den hautatu beharko dugu, balio lehenetsia hautatuko dugu, hau da, gakoa ez da inoiz baliogabetuko:

Please specify how long the key should be valid.

0 = key does not expire

<n> = key expires in n days

<n>w = key expires in n weeks

<n>m = key expires in n months

<n>y = key expires in n years

Key is valid for? (0)

Hurrengo pausoan erabiltzaile-izena, iruzkin bat eta e-posta helbide bat eskatuko dizkigu:

Real name: Asier Iturralde Sarasola

Email address: asier.iturralde@gmail.com

Comment: Hau iruzkin bat da.

You selected this USER-ID:

"Asier Iturralde Sarasola(Hau iruzkin bat da.)<asier.iturralde@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O

Ondoren, pasa-esaldi bat eskatuko digu gako sekretua babesteko:

Enter passphrase:

Pasa-esaldia idatzi ondoren, teklatuan ausazko zerbait idatzi, sagua mugitu, diskoak erabili, ... beharko dugu ausazko zenbaki sortzaileak behar adina entropia lortu arte. Bukatzean gako publiko eta pribatuak sortu dituela jakinaraziko digu. Egiaztatzeko gpg –list-key erabiliko dugu:

```
gpg --list-key asier.iturralde@gmail.com
```

pub 2048R/30736D81 2011-08-30

uid Asier Iturralde Sarasola (Hau iruzkin bat da.) <asier.iturralde@gmail.com>

sub 2048R/3AC0A2FA 2011-08-30

Gako publikoa testu fitxategi batetara esportatuko dugu:

```
gpg -a --export 30736D81 > fwknop-client.asc
```

Ondoren gako publikoa zerbitzarira igoko dugu SCP erabiliz:

```
scp fwknop-client.asc 192.168.56.2:
```

Zerbitzarian ere pauso berdinak jarraitu behar ditugu bere gako parea sortzeko, kasu honetan datu hauekin:

Real name: Asier Iturralde Sarasola

Email address: asier.iturralde@zerbitzaria.org

Comment: Hau zerbitzariko gako parea sortzeko da

You selected this USER-ID:

"Asier Iturralde Sarasola (Hau zerbitzariko gako parea sortzeko da) <asier.iturralde@zerbitzaria.org>"

Pasa-esaldia ezarriko dugu eta lehen bezala entropia sortuko dugu nahikoa dela esaten digun arte. Egiazta dezagun gako parea sortu duela:

```
gpg --list-key asier.iturralde@zerbitzaria.org
```

pub 2048R/B0256533 2011-08-30

uid Asier Iturralde Sarasola (Hau zerbitzariko gako parea sortzeko da) <asier.iturralde@zerbitzaria.org>

sub 2048R/98AC0AB0 2011-08-30

Bezeroan egin bezalaxe, zerbitzarian ere gako publikoa testu fitxategi batetara esportatuko dugu:

```
gpg -a --export B0256533 > fwknop-server.asc
```

Ondoren, bezerotik zerbitzariaren gako publikoa deskargatuko dugu SCP erabiliz:

```
scp 192.168.56.2:fwknop-server.asc .
```

Hurrengo pausoa gako publikoak gako sortara batzea eta sinatzea da. Lehenik bezeroan:

```
gpg --import fwknop-server.asc
```

gpg: key B0256533: public key "Asier Iturralde Sarasola (Hau zerbitzariko gako parea sortzeko da) <asier.iturralde@zerbitzaria.org>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

```
gpg --sign-key asier.iturralde@zerbitzaria.org
```

Ziur al gauden galdetuko digu, baietz erantzungo dugu:

Are you sure that you want to sign this key with your key "Asier Iturralde Sarasola (Hau iruzkin bat da.) <asier.iturralde@gmail.com>" (30736D81)

Really sign? (y/N) y

Ondoren, bezeroko gakoaren pasa-esaldia eskatuko digu sinatzeko:

You need a passphrase to unlock the secret key for user: "Asier Iturralde Sarasola (Hau iruzkin bat da.) <asier.iturralde@gmail.com>"

2048-bit RSA key, ID 30736D81, created 2011-08-30

Ondoren, zerbitzarian berdina egin behar da, lehenik scp bidez igo dugun bezeroaren gako publikoa inportatu:

```
gpg --import fwknop-client.asc
```

gpg: key 30736D81: public key "Asier Iturralde Sarasola (Hau iruzkin bat da.) <asier.iturralde@gmail.com>" imported

gpg: Total number processed: 1

gpg: imported: 1 (RSA: 1)

Ondoren, sinatu. Bezeroan bezala ziur al gauden galdetuko digu eta ondoren zerbitzariko gakoaren pasa-esaldia eskatuko digu.

```
gpg --sign-key asier.iturralde@gmail.com
```

Hurrengo pausoa fwknopd zerbitzaria konfiguratzea da, horretarako /etc/fwknop/access.conf fitxategia egokitu beharko dugu GPG autorizazioak erabiltzeko. KEY:<pasahitza> iruzkindu egingo dugu eta GPG_HOME_DIR, GPG_DECRYPT_ID, GPG_DECRYPT_PW eta GPG_REMOTE_ID

lerroei hasierako iruzkin ikurra (#) kendu eta dagozkien balioak idatziko ditugu:

 # See the fwknop man page for a comprehensive treatment of the various

 # access control variables. See the README.ACCESS file for additional

 # examples on how to configure access for various services.

#

###

#

 # $Id: access.conf 1145 2008-06-28 15:41:17Z mbr $

#

default Single Packet Authorization (SPA) via libpcap:

SOURCE: ANY;

OPEN_PORTS: tcp/22;   ### for ssh (change for access to other services)

#KEY: <pasahitza>;

FW_ACCESS_TIMEOUT: 30;

if you want to use GnuPG keys (recommended) then define the following

variables

GPG_HOME_DIR: /home/asier/.gnupg;   #gpg direktorioa, gpg –gen-key egikaritzean sortua

GPG_DECRYPT_ID: B0256533;   #Zerbitzariko gako publikoaren IDa

GPG_DECRYPT_PW: <zerbitzariko pasa-esaldia>;

GPG_REMOTE_ID: 30736D81;   #Bezeroko gako publikoaren IDa

fwknop-server martxan dagoela eta suhesia behar bezala konfiguratuta dagoela ziurtatu ondoren, bezerotik konektatzen saiatuko gara modu honetako komando batekin:

```
fwknop -A tcp/22 --gpg-recip SERVER_KEY --gpg-sign CLIENT_KEY -s
-k SERVER_IP
```

-s: Jatorriko IPari baimena eman behar zaiola adierazten du. Zerbitzariaren sare berean gaudenean erabiltzen da. Bezeroa eta zerbitzaria sare desberdinetan baleude eta bezeroa NAT suhesi baten azpian balego, -w erabili behar litzateke -s-ren ordez. Modu honetan www.whatismyip.com erabiliko luke bezeroak bere benetako IP-a erabiltzeko sare lokaleko IP-aren ordez.

```
fwknop -A tcp/22 --gpg-recip B0256533 --gpg-sign 30736D81 -s -k
192.168.56.2
```

[+] Starting fwknop client (SPA mode)...

[+] Enter the GnuPG password for signing key: 30736D81

Pasa-esaldia eskatuko digu:

GnuPG signing password:

[+] Building encrypted Single Packet Authorization (SPA) message...

[+] Packet fields:

Random data: 1097103355369989

Username: asier

Timestamp: 1314702002

Version: 1.9.12

Type: 1 (access mode)

Access: 0.0.0.0,tcp/22

SHA256 digest: htn2aC2ynu95j132FvZl8NQfDDLUfJzRKZlmxqwLwro

[+] Sending 996 byte message to 192.168.56.2 over udp/62201...

Eta orain ssh bidez konektatu ahalko gara:

```
ssh asier@192.168.56.2
```

Zerbitzarian iptables -L egikarituz bezeroaren IPtik SSH konexioa onartzeko suhesiaren arauak nola aldatu diren ikusiko dugu:

```
sudo iptables -L
```

Chain INPUT (policy ACCEPT)

target prot opt source destination

FWKNOP_INPUT all -- anywhere anywhere

ACCEPT all -- anywhere anywhere state RELATED,ESTABLISHED

DROP all -- anywhere anywhere

Chain FORWARD (policy ACCEPT)

target prot opt source destination

Chain OUTPUT (policy ACCEPT)

target prot opt source destination

Chain FWKNOP_INPUT (1 references)

target prot opt source destination

ACCEPT tcp -- 192.168.56.1 anywhere tcp dpt:ssh

5.2.1 Bibliografia

- https://help.ubuntu.com/community/SinglePacketAuthorization
- http://www.cipherdyne.org/fwknop/docs/gpghowto.html

5.3 Wake on LAN

5.3.1 Virtual Box-ekin saiakera

Lehen saiakera VirtualBox-eko Ubuntu 10.04 makina birtual batekin egin nahi nuen. Baina BIOSean aldaketak egiten saiatu nintzenean aurkitu nuen gauza bakarra F12 sakatu eta makina nondik abiarazi nahi dugun aukeratzeko menua da (CD-ROM-etik, disko gogorretik, ...).

Ikerketan ari nintzela komando lerrotik VirtualBox makina birtualekin lan egiteko komando ahaltsu bat aurkitu nuen: *vboxmanage*.

vboxmanage --help egikarituta egundoko aukera pila eskaintzen dituela ikusi nuen baina ez nuen aurkitu WOL-ekin erlazionatutako ezer.

5.3.2 Ubuntu instalatu USB pendrive batean

Nire eramangarriak, Sony Vaio VGN-N21Z/W, ez du eskaintzen Wake On LAN-erako aukerarik. (Edo eskaintzen badu nik ez dut aurkitu). Hori dela eta, lankide ohi baten ordenagailuak erabili ditugu probak egiteko. Arazo bat zegoen ordea, bi ordenagailu ditu baina biak Windows XP sistema eragiledunak. Ubuntu-rekin egin nahi genituen probak, beraz aukera bat Ubuntu bigarren sistema eragile bezala instalatzea zen. Bi ordenagailuen disko gogorrak nahiko beteta zeuden, edukien zati bat kanpoko disko gogor batetara pasa behar genuen, gainerakoen segurtasun kopia bat egin eta disko gogorra desfragmentatu. Baina orduan beste ideia bat bururatu zitzaigun, zergatik ez instalatu Ubuntu USB pendrive batean eta ordenagailua bertatik abiarazi. Modu honetan ordenagailuen disko gogorretan ez genuen aldaketarik egin beharko eta gainera sistema eragilearen eta beharrezko zerbitzarien eta aplikazioen instalazioa behin bakarrik egin beharko genuen.

Instalazio prozesua oso erraza da, beharrezko gauzak ordenagailu bat, USB pendrive bat eta Ubuntu-ren LiveCD bat dira.

1. Behar izanez gero BIOS-ean aldaketak egin abiarazteko garaian CD-ROM-ak disko gogorrak baina lehentasun gehiago izan dezan.
2. Ordenagailua piztu USB pendrive-a konektatuta duela eta Ubuntu LiveCD bat abiarazi.
3. Ubuntu instalatu nahi dugula aukeratu. Jarraitu beharreko prozedura ohiko instalazioetakoa bezalakoxea da, desberdintasun txiki batekin, instalazioa disko gogorrean egin ordez USB pendrivean (orokorrean sdb) egin nahi dugula adierazi behar dugu non instalatu nahi dugun galdetzean.

Emaitzarekin nahiko konforme gelditu nintzen, disko gogorrean instalatutako Ubuntu baina pixka bat mantsoagoa da baina nahiko ondo erantzuten du. Alderik handiena abiarazteko denboran dago, disko gogorretik abiaraztean baino nabarmen mantsoagoa da, hala ere LiveCD bat baina erosoagoa da eta azkarrago abiarazten da, gainera nahi ditugun aplikazioak edo zerbitzariak instala ditzakegu edozein ordenagailutan erabili ahal izateko Ubuntu instalatu beharrik gabe. Horretarako ordenagailuan egin beharreko aldaketa bakarra abiarazteko lehentasunak aldatzea da.

5.3.3 BIOS-a egokitu USB-tik abiarazteko

Ubuntu pendrive-ean instalatu ondoren, ordenagailua USBtik abiaraztea lortu behar dugu, horretarako ondokoa egin beharko dugu:

1. BIOS-aren konfiguraziora sartu DEL sakatuz
2. Advanced BIOS Features hautatu

3. Bertan abiarazterakoan nahi ditugun lehentasunak hautatu:

   First Boot Device: [USB-HDD] (Orain arteko aukera [Floppy] zen.)

   Second Boot Device: [CDROM]

   Third Boot Device: [Hard Disk]
4. F10 sakatu aldaketak gordetzeko.

Prozedura hau BIOS batetik bestera aldakorra den arren gutxi gora behera antzeko zerbait izango da.

5.3.4 BIOS-a egokitu WOL erabiltzeko

BIOS desberdin asko daude eta bakoitzean modu desberdin batean aktibatzen da WOLerako aukera. Orokorrean "Power Management" atalean egon behar luke. Edo NIC-a PCIari loturik badago "PCI peripherals" edo antzeko beste izen bat duen atalean egon behar luke.

Probak egiteko erabilitako ordenagailuetako batean BIOSera sartu eta ondokoa egin behar da:

Power > APM Configuration > Power On By PCI/PCIE Device -> Enabled

Aurretik esan bezala nire eramangarriaren BIOSean ez dut aurkitu WOL erabiltzeko modurik ematen duen aukerarik.

Probak egiteko erabilitako beste ordenagailuan alderantzizkoa gertatzen da, bere BIOSean ez dut aurkitu WOL desaktibatzeko modurik.

5.3.5 Sistema eragilea egokitu

BIOSean Wake on LAN aktibatu ondoren sistema eragileari jakinarazi behar diogu itzali aurretik, bestela makina ez da piztuko.

NIC-ak WOL onartzen duen jakiteko ethtool erabil daiteke:

```
sudo ethtool eth0
```

Settings for eth0:

...

Supports Wake-on: pg

Wake-on: d

...

Support Wake-on-ek "g" badauka pakete magiko bidezko WOL erabil daiteke.

Ondoko komandoa exekutatu behar da itzali aurretik WOL bidez piztu nahi dugun bakoitzean:

```
sudo ethtool -s eth0 wol g
```

Egiazta dezagun WOL erabilgarri dagoela orain (Wake-on aukeran: g agertu behar du):

```
sudo ethtool eth0
```

Settings for eth0:

...

Wake-on: g

...

5.3.6 DD-WRT-ren instalazioa D-Link DIR-320 bideratzailean

DD-WRT firmware-ak jatorrizko firmwarea baino aurreratuagoa da eta aukera gehiago eskaintzen ditu, batez ere Wake On LAN erabiltzeko aukera ematen duelako aukeratu dut.

1. http://www.dd-wrt.com/site/support/router-database webgunera nabigatu eta bertan gure bideratzailearen izena idatziko dugu: D-Link DIR-320.

   Ondoko datuak agertuko zaizkigu:

Manufacturer	*Model*	*Revision*	*Supported*	*Activation required*
D-Link	*DIR-320*	*A1/A2*	*yes*	*no*

2. Klik egitean gure bideratzailearen hainbat datu agertuko zaizkigu, chipset-a, RAM memoria kopurua, FLASH memoria kopurua, DD-WRT-ren zein bertsio erabil ditzakegun eta flasheatzeko gida:

   http://www.shadowandy.net/2008/06/flashing-guide-for-dir-320.htm

   Gomendatzen zaigun dd-wrt-ren bertsioa ondokoa da:

   v24 preSP2 [Beta] Build 14896

3. Aurrera jarraitu aurretik gure makinaren IP-a 192.168.0.10 izatea lortu behar dugu (Sare maskara: 255.255.255.0), horretarako:
   1. Egin klik eskuineko botoiarekin NetworkManager Applet-aren gainean.
   2. **Editatu konexioak...** klikatu
   3. Hautatu gure sareko konexioa eta sakatu **Editatu**
   4. IPv4 fitxara joan eta bertan ondoko aukerak ezarri:
      - **Metodoa**: Eskuz
      - **Helbidea**: 192.168.0.10
      - **Sare-maskara**: 255.255.255.0
      - **Atebidea**: 192.168.0.1
      - **DNS zerbitzariak**: 192.168.0.1
   5. **Aplikatu...** sakatu
   6. Pasahitza idatzi eta sakatu **Autentifikatu**
   7. ifconfig bidez gure IP helbide lokala 192.168.0.10 dela egiaztatu
4. tftp paketea instalatu behar dugu (*Trivial File Transfer Protocol* bezeroa):

   ```
   sudo apt-get install tftp
   ```

   Trivial File Transfer Protocol (TFTP) fitxategi transferentziarako protokolo sinple bat da.

FTPren aldean oso mugatua da eta ez da batere segurua, ez baitu autentifikaziorik eskaintzen. Abantaila nagusia oso memoria gutxi erabiltzen duela da eta erabilera nagusia bideratzaile eta antzeko gailu mugatuetara fitxategiak transferitzea da. Segurtasun kaskarra dela eta sare lokaletan soilik erabiltzea gomendatzen da.

5. Instalatu nahi dugun dd-wrt-ren binarioa deskargatu behar dugu, mini generic erabiliko dugu: http://www.dd-wrt.com/routerdb/de/download/D-Link/DIR-320/A1/A2/dd-wrt.v24_mini_generic.bin/3750
6. Fitxategia deskargatu ondoren firmware.bin bezala berrizendatuko dugu.
7. Terminal bat ireki eta ondokoa idatzi, baina azken lerroa idatzi ondoren ez sakatu enter:

```
cd /home/asier/Deskargak/
~/Deskargak$ tftp
tftp> binary
tftp> trace
```

Packet tracing on.

```
tftp> rexmt 1
tftp> connect 192.168.0.1
tftp> put firmware.bin
```

8. Bideratzailearen entxufea atera, bideratzailea internetera konektatzen duen kablea kendu eta gure makina eta bideratzailea ethernet kable bidez konektatuta daudela egiaztatu.
9. Bideratzailea entxufatu eta 2 segundo itxaron ondoren ENTER sakatu. Gauzak ondo badoaz tftp lanean hasiko da, bideratzailera firmware berria bidaltzen. Bestela 7. pausora itzuli.
10. 10 minutu inguru itxaron ondoren Network Manager appleta erabiliz konexioaren konfigurazioa aldatuko dugu:
    - **Metodoa**: Helbide automatikoak soilik (DHCP)
    - **DNS zerbitzariak**: 192.168.1.1
11. **Aplikatu...** sakatuko dugu azkenik.
12. Nabigatzaile bat ireki eta http://192.168.1.1 orrira joango gara. Bertan dd-wrt-ren kontrol panela agertuko zaigu eta erabiltzaile izen eta pasahitz berriak idatzi beharko ditugu, momentuz erabiltzen dituen balio lehenetsiak (Erabiltzaile-izena: root, Pasahitza: admin) ez baitira seguruak.
13. Ondoren, sakatu **Change Password**.
14. Interneterako konexioa konfiguratuko dugu ondoren, nire kasuan PPPOE bidezko modem bat erabiliko dudanez Connection Type: PPPOE hautatuko dut:

    Setup -> WAN Setup -> WAN Connection Type

    - **Connection Type**: PPPOE
    - Modemaren erabiltzaile-izena eta pasahitza idatzi
15. **Apply Settings** sakatu.

5.3.7 Dyndns zerbitzuan kontu bat ireki eta bideratzailea konfiguratu erabiltzeko

1. http://dyn.com/dns/dyndns-free/ webgunera joan eta GET IT NOW sakatu
2. https://www.dyndns.com/account/services/hosts/add.html webgunea agertuko zaigu, bertan ondoko datuak bete behar ditugu:
   - **Hostname**: Erabili nahi dugun helbidea, adibidez, adibidea.dyndns.org
   - **Service Type**: Host with IP address
   - **IP Address**: Gure uneko IP publikoa
4. **Add To Cart** botoia sakatu
5. Kontu bat sortu behar dugu, horretarako datu hauek bete beharko ditugu: Erabiltzaile-izena, pasahitza eta e-posta helbide bat.
6. **Create Account** sakatu
7. Idatzi dugun e-posta kontura sartu eta dyndns-tik bidali diguten e-postako konfirmaziorako loturan klik egin behar da aktibatzeko.
8. adibidea.dyndns.org ostalari izena erabil dezakegu hemendik aurrera gure IP dinamikoa ordezkatzeko, baina lehenik bideratzaileari jakinarazi behar dizkiogu gure asmoak.
9. Nabigatzailean 192.168.1.1 helbidera nabigatu.
10. **Setup > DDNS > DDNS Service** orrira joan eta DynDNS hautatu.
11. Erabiltzaile-izena, pasahitza, ostalari-izena (adibidea.dyndns.org) eta mota (Dinamikoa) hautatu eta **Apply Settings** sakatu.

5.3.8 Ataka birbidalketa konfiguratu bideratzailean

Bi ataka birbidalketa arau konfiguratuko ditugu bideratzailean, bat WOL pakete magikoentzat eta bestea SSH-rentzat. Has gaitezen Wake on LAN-ekin:

1. Nabigatzailean 192.168.1.1 helbidera nabigatu.
2. **NAT/QoS** fitxa hautatu, pasahitza eskatuko zaigu.
3. **Forwards** atalean **Add** sakatu.
4. Datuak:
   - **Application**: WOL
   - **Port from**: 7 (Bideratzailearen ataka)
   - **Protocol**: Both (TCP eta UDP biak, badaezpada ere)
   - **IP Address**: 192.168.1.130 (Zerbitzariaren IP pribatua)
   - **Port To**: 7 (Zerbitzariaren ataka)
   - **Enable**: Bai
5. **Apply Settings** sakatu.

Berdin egingo dugu SSH-rekin, baina kasu honetan:

- **Application**: SSH
- **Port from**: 2222
- **Protocol**: TCP
- **Port To**: 22

VNC eta NX zerbitzuentzat ez dugu ataka birbidalketarik konfiguratuko, SSH tuneletan zehar konektatuko baikara.

5.3.9 Bideratzailearen konfigurazioa Wake On LAN onartzeko

Aurreko atalean beharrezko ataka birbidalketa araua konfiguratu ondoren ondoko pausoak eman behar dira Wake On LAN gaitzeko:

1. Nabigatzailean 192.168.1.1 helbidera nabigatu.
2. **Administration** > **WOL** orrira joan eta bertako **Available Hosts** zerrendan hautatu gure makinari dagokion lerroa eta **Enable WOL** onartu.
3. **WOL Addresses**-en agertuko lerro berri bat agertuko zaigu:

MAC Address	*Host Name*	*Net Broadcast*
A1:B2:C3:D4:E5:F6	*ubuntuUSB*	*192.168.1.255*

4. **Apply Settings** sakatu
5. ARP arau estatiko bat sortu gehitu behar dugu ondoren, horretarako **Administration** > **Commands** orrira joango gara eta ondoko bi lerroak gehitu ondoren **Save Startup** sakatuko dugu:

   ```
   ip neigh change 192.168.1.130 lladdr ff:ff:ff:ff:ff:ff nud permanent dev br0
   ```

   ```
   ip neigh add 192.168.1.130 lladdr ff:ff:ff:ff:ff:ff nud permanent dev br0
   ```

6. Azkenik, bideratzailea berrabiarazi behar dugu, horretarako **Administration** > **Management** orrira joan eta **Reboot router** sakatuko dugu.

5.3.10 gWakeOnLan-ekin proba egin

1. Makina bat WOLerako prestatu ondoren beste makinan gWakeOnLan abiarazi:

   ```
   Aplikazioak > Internet > gWakeOnLan
   ```

2. Piztu nahi dugun makinaren IP-a eta MAC helbidea idatzi eta pakete magikoa bidaltzeko botoia sakatu ondoren behar bezala piztu da.

   Proba sare lokalean eta interneten zehar egin dut (dyndns erabiliz).

5.3.11 Bibliografia

- http://code.google.com/p/gwakeonlan/wiki/Enable_WOL_Bios
- http://code.google.com/p/gwakeonlan/wiki/Enable_WOL_OS
- http://www.shadowandy.net/2008/06/flashing-guide-for-dir-320.htm
- http://en.wikipedia.org/wiki/Trivial_File_Transfer_Protocol
- http://www.dd-wrt.com/wiki/index.php/Port_Forwarding
- http://www.dd-wrt.com/wiki/index.php/WOL

5.4 SSH

5.4.1 OpenSSH zerbitzaria instalatu eta konfiguratzea

OpenSSH zerbitzaria instalatzeko ondoko komandoa egikarituko dugu:

```
sudo apt-get install openssh-server
```

/etc/ssh/sshd_config fitxategiaren bidez konfiguratzen da. Root bezala saioa hasteko aukera kentzea komeni da, horretarako:

PermitRootLogin no

Eta asier erabiltzaileari sarrera onartu:

AllowUsers asier

Zerbitzaria abiarazteko:

```
sudo /etc/init.d/ssh start
```

Eta gelditzeko:

```
sudo /etc/init.d/ssh stop
```

Proba egiteko bezerotik zerbitzarira konektatuko naiz interneten zehar. Zerbitzariaren IPa dinamikoa denez lehen sortu dugun dyndns helbidea erabiliko dugu konexioa ezartzeko:

```
ssh -p 2222 asier@adibidea.dyndns.org
```

Egiaztatzeko netstat erabiliko dut, lehenik bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:38669 74.125.230.210:2222 ESTABLISHED 3814/ssh

Eta ondoren zerbitzarian:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.130:22 192.168.1.1:38669 ESTABLISHED 1708/sshd: asier [p

5.4.2 Gako publiko bidezko autentifikazioa

SSH saioaren segurtasun maila handitzeko edota pasahitzik gabe autentifikatzeko SSHren gako publiko bidezko kriptografia erabil daiteke. Pasahitzik gabeko autentifikazioa interesgarria da script batean SSH edo SCP erabili behar dugun kasuetarako adibidez. Kasu hauetan urruneko makinan saio bat hastean erabiltzailearen pasahitza idatzi ordez SSH-k erronka-erantzun protokoloa abiaraziko du, zerbitzariko enkriptatutako gako publikoa eta bezeroaren gako pribatua erabiliz. Modu honetan, ez dago sarean zehar pasahitza bezalako informazio sentikorrik bidali beharrik.

Gako publiko/pribatu pare bat sortzeko ondoko komandoa erabili behar da makina lokalean, pasa-esaldi bat eskatuko digu eta gako publikoa non gordetzea nahi dugun galdetuko digu.

```
ssh-keygen -t dsa
```

Pasa-esaldia erabiltzea ez da derrigorrezkoa baina erabiltzen ez badugu makina lokaleko gure kontura sarrera lortzen duen edozeinek gure gako publikoa ezagutzen duten urruneko makina guztietara sarbidea lortuko du.

Ondoren gure gako publikoa urruneko makinara kopiatu behar dugu, ondoko komandoa erabiliz:

```
ssh-copy-id -i ~/.ssh/id_dsa.pub remoteuser@remotebox
```

Nire kasuan:

```
ssh-copy-id -i /home/asier/.ssh/id_dsa.pub asier@192.168.56.2
```

Ondoko mezua jaso dut orduan:

Now try logging into the machine, with "ssh 'asier@192.168.56.2'", and check in:

.ssh/authorized_keys to make sure we haven't added extra keys that you weren't expecting.

Jaso dugun mezuak iradoki bezala makina lokaleko gako publikoa (/home/asier/.ssh/id_dsa.pub) eta urruneko makinara kopiatu duguna bat datozela egiaztatzea komeni da (/home/asier/.ssh/authorized_keys).

Orain ssh bidez saio bat hasten saiatzen naizenean mezu bat agertzen zait, gako pribatua desblokeatzeko pasahitza eskatzen didana, ez dakit zergatik.

Gako publiko berria sortu dut eta orain ez zait lehengo mezua agertzen baina hala ere ezin dut gako publikoa erabiliz konektatu:

```
ssh asier@192.168.56.2
```

Agent admitted failure to sign using the key.

asier@192.168.56.2's password:

Ondoko komandoa exekutatu ondoren lortu dut:

```
ssh-add
```

Enter passphrase for /home/asier/.ssh/id_dsa:

Identity added: /home/asier/.ssh/id_dsa (/home/asier/.ssh/id_dsa)

Eta orain bai SSH bidez konektatu naiteke gako publiko bidez autentifikatuz.

5.4.3 ~/.ssh/config fitxategiaren konfigurazioa

IP helbideen ordez ezizen bat erabil daiteke, horretarako OpenSSH bezeroaren konfigurazio fitxategia (~/.ssh/config) editatu behar da. Ostalariaren izena (urrun) eta IP zenbakia gehituko ditugu, honela:

Host urrun

HostName 192.168.56.2

Modu honetan "urrun" idatz dezakegu 192.168.56.2-ren ordez, honek gauzak asko errazten dituelarik.

Orain ssh bidez 192.168.56.2 urruneko makinara konektatzeko nahikoa da ondokoa idaztea:

```
ssh asier@urrun
```

Beti erabiltzaile bera erabiltzen badugu gauzak oraindik gehiago sinplifika ditzakegu erabiltzaile izena konfigurazio fitxategian gehituz:

Host urrun

HostName 192.168.56.2

User asier

Orain konektatzeko nahikoa da ondokoa erabiltzea:

```
ssh urrun
```

Konfigurazio fitxategian nahi adina ostalari desberdin gehi daitezke.

5.4.4 GUI programen birbidaltzea SSH bidez

Aplikazio grafikoak SSH bidez birbidaltzea oso erraza da, nahikoa da -X aukera gehitzea. Adibidez, demagun gedit erabili nahi dugula urruneko ordenagailu batean:

```
ssh -X asier@192.168.56.2
```

Eta ondoren:

```
gedit &
```

Programa modu arruntean abiarazten da, desberdintasun bakarra modu lokalean exekutatzean baino pixka bat mantsoago dabilela da. Bukaerako "&"-ak atzealdean martxan jarriko dugula esan nahi du, hau da, terminalean komandoak idazten jarraitu ahal izango dugula, aplikazioa bukatu arte itxaron beharrik gabe.

Gauza bera lor daiteke komando bakarra erabiliz:

```
ssh -f -X asier@192.168.56.2 gedit
```

-f: atzealdean exekutatu dadin

Aplikazio bat abiarazten saiatzean pantaila ezin duela aurkitu esaten badigu, Main software biltegiko xauth falta dela izan daiteke arrazoia. Xauth mahaigaineko banaketetan ohiko aplikazio bat den arren ez baita hala izaten zerbitzarietan.

Banda zabalera kaskarra dela eta aplikazioak mantso doazela irudituz gero interesgarria izan daiteke -C aukera erabiltzea SSH konpresioa aktibatzeko, adibidez:

```
ssh -fXC asier@192.168.56.2 gedit
```

Arazoen konponketa

Ataka birbidalketa egiten saiatzean honelako mezu bat jasoz gero:

bind: Address already in use

channel_setup_fwd_listener: cannot listen to port: <port number>

Could not request local forwarding.

Ataka horretan dagoeneko norbait entzuten dagoela esan nahi du. Ezingo dugu ataka horretan entzun beste erabiltzaileak bukatu arte.

Ataka birbidalketa ez badabil eta ez badugu ohar mezurik jasotzen, arrazoia SSH zerbitzariak birbidalketa desgaituta egotea izan daiteke.

Exekutatu hurrengo komandoa zerbitzarian:

```
grep Forwarding /etc/ssh/sshd_config
```

...

X11Forwarding no

AllowTcpForwarding no

...

Aurreko bi lerroetan "no" jartzen badu birbidalketa desgaituta dago eta "yes"-ekin ordezkatu behar genituzke.

5.4.5 Nabigazio segurua Firefox-ekin SSH tunel bidez proxy zerbitzari bat erabiliz

SSH tunela abiaraziko dugu makina lokaletik proxy bezala erabiliko dugun makinara:

```
ssh -ND 9999 asier@192.168.1.130
```

Pasahitza eskatuko digu eta ondoren zain gelditu ko da, -N erabili dugunez ez baitu prompt interaktiborik abiatuko.

Netstat erabiliko dugu tunela abiarazi dugula egiaztatzeko, lehenik makina lokalean:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:47072 192.168.1.130:22 ESTABLISHED 2229/ssh

Eta ondoren, urruneko makinan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.130:22 192.168.1.139:47072 ESTABLISHED 1997/sshd: asier [p

Proxy-a martxan jarri eta gero, Firefox konfiguratu behar dugu proxy-a erabil dezan zuzenean konektatu ordez. Horretarako,

1. "Editatu" menuko "Hobespenak"en klik egin.
2. "Aurreratua"n klik egin eta hautatu bertako "Sarea" fitxa.
3. "Konfiguratu Firefox Internetera nola konektatzen den" aukeraren ondoan dagoen "Ezarpenak..." botoian klik egin.
4. "Eskuzko proxy-konfigurazioa:" aukeratu eta bertan datu hauek idatzi:
   - SOCKS ostalaria: localhost
   - Ataka: 9999
5. Ados sakatu

Proba egiteko http://www.google.com erabiliko dut. Bertara nabigatu eta netstat egikaritu dut makina lokalean eta urruneko proxy zerbitzarian:

Makina lokalean:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:47072 192.168.1.130:22 ESTABLISHED 2229/ssh

tcp 0 0 127.0.0.1:34059 127.0.0.1:9999 ESTABLISHED 2461/firefox-bin

tcp 0 0 127.0.0.1:34060 127.0.0.1:9999 ESTABLISHED 2461/firefox-bin

tcp 0 0 127.0.0.1:9999 127.0.0.1:34059 ESTABLISHED 2229/ssh

tcp 0 0 127.0.0.1:9999 127.0.0.1:34060 ESTABLISHED 2229/ssh

Proxy zerbitzarian:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.130:52500 209.85.148.103:80 ESTABLISHED 2065/sshd: asier

tcp 0 0 192.168.1.130:22 192.168.1.139:47072 ESTABLISHED 1997/sshd: asier [p

tcp 0 0 192.168.1.130:52625 209.85.148.105:80 ESTABLISHED 2065/sshd: asier

Ikusten denez konexio guztiak SSH tunela zeharkatu eta proxy zerbitzarian zehar bideratzen dira internetera.

Proxy bidezko nabigazioa bukatzean hasieran ireki dugun SSH tunela itxiko dugu Ctrl+C sakatuz. Ondoren, Firefox-en konexio-ezarpenetan berriz ere "Proxy-rik ez" hautatuko dugu.

5.4.6 SCP

Makina lokaletik urruneko makina batera edo alderantziz fitxategiak modu seguruan kopiatzeko scp komandoa erabil daiteke.

5.4.6.1 Fitxategi bakar bat kopiatzea

```
scp fitxategia.txt erabiltzailea@urrunekoMakina:/katalogoa
```

non

"fitxategia.txt" urruneko makinara kopiatu nahi dugun fitxategia den,

"erabiltzailea" urruneko makinako erabiltzaile izena,

"urrunekoMakina" urruneko makinaren izena edo IP helbidea eta

"/katalogoa" fitxategia kopiatu nahi dugun urruneko makinako katalogoa.

Adibidez,

```
scp /home/asier/proba.txt asier@192.168.56.2:/home/asier/
```

asier@192.168.56.2's password:

proba.txt 100% 0 0.0KB/s 00:00

Egiazta dezagun urruneko makinan jaso dugula fitxategia:

```
ls /home/asier/ | grep proba
```

proba.txt

scp komandoak SSH tunel bat irekitzen du urruneko makinara fitxategia kopiatzeko eta bukatzean itxi.

cp komando arruntak bezala fitxategiari izena aldatzeko aukera eskaintzen du, nahikoa da izen berria urruneko katalogoaren izenari gehitzea.

Alderantzizkoa, hau da, urruneko makinatik makina lokalera fitxategi bat kopiatzeko berriz ondoko sintaxia erabiltzen da:

```
scp erabiltzailea@urrunekoMakina:/katalogoa/fitxategia /katalogo/lokala
```

non /katalogo/lokala fitxategia kopiatu nahi dugun makina lokaleko katalogoa den.

Adibidez, urruneko makina fitxategi bat sortuko dugu makina lokalera kopiatzeko:

```
touch /home/asier/proba2.txt
```

Orain makina lokaletik:

```
scp asier@192.168.56.2:/home/asier/proba2.txt /home/asier/
```

asier@192.168.56.2's password:

proba2.txt 100% 0 0.0KB/s 00:00

Egiazta dezagun:

```
ls /home/asier/ | grep proba
```

proba2.txt

proba.txt

5.4.6.2 Katalogoak kopiatzea

Katalogo oso bat bere eduki eta guzti kopiatzeko -r gehitu behar zaio komandoari.

Honela, katalogo bat makina lokaletik urruneko makinara kopiatzeko:

```
scp -r /katalogo/lokala erabiltzailea@urrunekoMakina:/urruneko/katalogoa
```

Eta alderantziz, urruneko makinatik makina lokalera kopiatzeko:

```
scp -r erabiltzailea@urrunekoMakina:/urruneko/katalogoa /katalogo/lokala
```

5.4.7 Bibliografia

- http://principialabs.com/beginning-ssh-on-ubuntu/
- http://support.suso.com/supki/SSH_Tutorial_for_Linux
- https://help.ubuntu.com/community/SSH/OpenSSH/PortForwarding
- http://lifehacker.com/237227/geek-to-live--encrypt-your-web-browsing-session-with-an-ssh-socks-proxy
- http://ubuntuforums.org/showthread.php?t=723025
- http://www.hypexr.org/linux_scp_help.php

5.5 VNC

5.5.1 VNC konexioa Vino (zerbitzaria) eta Vinagre (bezeroa) erabiliz

1. VNC zerbitzari bezala lan egingo duen makinan Vino abiaraziko dugu:

   Sistema > Hobespenak > Urruneko mahaigaina

2. **Partekatzea** atalean:

   - **Baimendu beste erabiltzaileek zure mahaigaina ikustea** (Balio lehenetsia ez da. Gure mahaigaina partekatu nahi badugu hautatu. Gainerako aukera guztiak desaktibatzen dira hautatuta ez dagoenean)
   - **Baimendu beste erabiltzaileek zure mahaigaina kontrolatzea** (Balio lehenetsia bai da. Urruneko erabiltzaileari gure teklatua eta sagua kontrolatzeko aukera eman nahi badiogu hautatu)

3. Ondoko mezua agertu zait mahaigaina ikusteko baimena ematean:

   Zure mahaigaina sare lokaletik soilik atzi daiteke. Besteek 192.168.1.130 edo ubuntuUSB.local helbidea erabiliz atzi dezakete zure ordenagailua.

4. **Segurtasuna** atalean ondoko aukerak daude:

   - **Ordenagailu honetarako sarbide bakoitza berretsi behar duzu** (Balio lehenetsia bai da. Nahigabeko sarrerak saihesteko, sarrerako konexio bakoitza onartu edo ukatzeko galdetuko zaigu.)
   - **Behartu erabiltzaileak pasahitza sartzera:** _____ (Balio lehenetsia ez da. Hautatu sarbidea pasahitzez babestu nahi badugu.)
   - **Konfiguratu sarea automatikoki konexioak onartzeko** (Balio lehenetsia ez da. Bideratzaileak UPnP eginbidea gaituta eduki behar du.)

5. **Jakinarazpen-area** atalean ondoko aukerak daude (Hiruetako bat hautatu behar da), ikonoa ze kasutan bistaratzea nahi dugun zehazteko dira:

   - **Beti erakutsi ikonoa**
   - **Bistaratu ikonoa norbait konektatzean soilik** (Balio lehenetsia)
   - **Inoiz ez bistaratu ikonoa**

6. Vinagre ireki, 192.168.1.130 edo ubuntuUSB.local zehaztu ostalari bezala eta konektatzen saiatuko gara.

   Zerbitzarian ondoko mezua agertu zait:

   Beste erabiltzaile bat zure mahaigaina ikusten saiatzen ari da.

   'asier-laptop' ordenagailuko erabiltzaile bat zure mahaigaina urrunetik ikusten edo kontrolatzen saiatzen ari da.

   Baimentzea nahi duzu hori egiteko?

   Debekatu Baimendu

7. **Baimendu** sakatzean Vinagre Urruneko Mahaigainen ikustailean VNC zerbitzariaren mahaigaina agertu zait. Zerbitzariko goiko panelean ikono bat agertu zait pertsona bat konektatu dela adierazteko.

Erabilera konkretu batzuetarako ondo dagoen arren, adibidez, arazoren batekin lagun gaitzan lagun batekin mahaigaina partekatzeko, nahiko mugatua da. VNC zerbitzari ahaltsuago bat probatuko dut ondoren: X11vnc.

5.5.2 X11vnc zerbitzariaren instalazioa

X11vnc zerbitzaria instalatzeko ondoko komandoa egikaritu behar dugu:

```
sudo apt-get install x11vnc vnc-java
```

Ondoren pasahitza zehaztu behar dugu:

```
x11vnc -storepasswd
```

Enter VNC password:

Verify password:

Write password to /home/asier/.vnc/passwd? [y]/n y

Password written to: /home/asier/.vnc/passwd

Suhesirik balego 5800 eta 5900 atakak irekita daudela ziurtatu. Zerbitzaria abiarazteko ondokoa exekutatu behar da:

```
x11vnc -forever -usepw -httpdir /usr/share/vnc-java/ -httpport 5800
```

Ordenagailua abiarazten den bakoitzean zerbitzaria martxan jartzea nahi badugu abioko aplikazioei gehitu behar diegu:

Sistema > Hobespenak > Abioko aplikazioak

5.5.3 Bezerotik zerbitzarira konektatzea

VNC zerbitzua eskuragarri dagoela egiaztatuko dugu lehenik:

```
nmap 192.168.0.101
```

Starting Nmap 5.00 (http://nmap.org) at 2011-08-22 17:28 CEST

Interesting ports on 192.168.0.101:

Not shown: 998 closed ports

PORT STATE SERVICE

5800/tcp open vnc-http

5900/tcp open vnc

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds

Ondoren Urruneko mahaigainen ikustailea abiaraziko dugu:

Aplikazioak > Internet > Urruneko mahaigainen ikustailea

Mezu hau agertu zait:

Menuen bizkortzaile eta lasterbideei buruz

Lehenetsi bezala, Vinagre menuen bizkortzaile eta lasterbideak desgaituta dituela etortzen da. Arrazoi nagusia programak sakatzen diren teklak ez interpretatzea da, tekla horiek urruneko ordenagailura bidaltzeko xedea baitu.

Portaera hau alda dezakezu hobespenen elkarrizketa-koadroa erabiliz. Informazio gehiagorako, irakurri dokumentazioa.

Mezu hau behin bakarrik agertuko da.

Urruneko mahaigain ikustailean ondoko aukerak hautatu behar ditugu:

- **Protokoloa**: gure kasuan VNC
- **Ostalaria**: 192.168.0.101 (ostalariaren IPa)
  - **Bilatu** botoiaren bidez VNC zerbitzariak bilatzeko aukera eskaintzen du, baina ez dut lortu sare lokaleko VNC zerbitzaria aurkitzerik.
- **Konexioa aukerak**
  - **Pantaila osoa**: Ez (Nahi izanez gero gure makinaren pantaila osoa erabil dezakegu ostalariko mahaigaina bistaratzeko)
- **VNC aukerak**
  - **Ikusteko soilik**: Ez (Baietz hautatuz gero ezingo ditugu sagua eta teklatua erabili)
  - **Eskalatzea**: Bai (Urruneko mahaigainaren tamaina gure pantailaren tamainara egokitu dadin)
    - **Mantendu aspektu-erlazioa**: Bai (Eskalatzean aspektu-erlazioa mantentzeko)
  - **Erabili JPEG konpresioak**: Ez (Bai hautatuta konpresioa erabiliko da konexioa azkartzeko)
  - **Kolore-sakonera**: Erabili zerbitzariaren ezarpenak. (Ikusiko dugun mahaigainaren irudiaren kalitatea zehazten da hemen, beste aukerak: Benetako kolorea (24 bit), Kolore altua (16 bit), Kolore baxua (8 bit) eta Kolore oso baxua (3 bit)).
  - **Erabili ostalaria _____ SSH tunel bat bezala** (SSH tunel bat erabiltzeko aukera eskaintzen du, aurrerago erabiliko dugu).

Nahi ditugun aukerak hautatu ondoren **Konektatu** sakatu, VNC pasahitza zehaztu eta ostalarira konektatuta gaude.

Egiaztatzeko netstat -tanp egikarituko dut makina lokalean:

```
netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.0.100:53912 192.168.0.101:5900 ESTABLISHED 2236/vinagre

Berdina egingo dugu urruneko makinan:

```
netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.0.101:5900 192.168.0.100:53912 ESTABLISHED 1637/x11vnc

5.5.4 VNCren SSH bidezko ataka birbidalketa

Segurtasun gehiago nahi badugu VNC zerbitzaria SSH bidez birbidaliko dugu, modu honetan sarean zehar egiten dugun konexio bakarra SSH bidezkoa izango da eta beraz gure datuak zifraturik bidaliko ditugu. Proba egin aurretik urruneko ordenagailuan SSH eta VNC zerbitzariak martxan daudela egiaztatu behar dugu. Urruneko makinako suhesiak behar bezala konfiguratuta egon behar du SSH zerbitzaria makina lokaletik eskuragarri egon dadin, VNC zerbitzariak ez du zertan eskuragarri egon, SSH bidez birbidaliko baitugu.

Urruneko konputagailura SSH bidez modu seguruan konektatuko gara eta VNC zerbitzariaren bidez mahaigain partekatua birbidaliko dugu gure ordenagailura.

```
ssh -L 5900:localhost:5900 asier@192.168.0.101
```

Ondoren, urruneko mahaigain ikustailean gure makina lokaleko 5900 atakara konektatuko gara birbidali dugun mahaigaina ikusteko. Horretarako, ostalari bezala localhost idatziko dugu.

Localhostera birbidali dugun urruneko makinako VNC zerbitzariaren pasahitza eskatuko digu, idatzi eta konektatuta gaude.

Egiaztatzeko netstat -tanp egikarituko dut makina lokalean:

```
netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.0.100:42473 192.168.0.101:22 ESTABLISHED 1956/ssh

tcp 0 0 127.0.0.1:57825 127.0.0.1:5900 ESTABLISHED 1960/vinagre

tcp 0 0 127.0.0.1:5900 127.0.0.1:57825 ESTABLISHED 1956/ssh

Ikusten denez, urruneko makinara ezarritako konexio bakarra SSH bidezkoa da, birbidalitako VNC zerbitzarira localhost bidez konektatu gara.

Berdina egingo dugu urruneko makinan:

```
netstat -tan | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:5900 127.0.0.1:33300 ESTABLISHED

tcp 0 0 192.168.0.101:22 192.168.0.100:42473 ESTABLISHED

tcp 0 0 127.0.0.1:33300 127.0.0.1:5900 ESTABLISHED

Kasu honetan ere emaitza berdina da, makina lokal eta urrunekoaren arteko konexio bakarra SSH bidezkoa da, VNC zerbitzaria localhost-ean dago eta SSH bidez birbidaltzen da.

5.5.4.1 VNCren SSH bidezko ataka birbidalketa Vinagre Urruneko Mahaigainen Ikustailea erabiliz

Urruneko mahaigain ikustaileak prozesu hau automatizatzeko aukera eskaintzen du. Horretarako, aukera hauek zehaztu behar dira:

Ostalaria: localhost

Erabili ostalaria 192,168.0.101 **SSH tunel bat bezala**.

Eta **Konektatu** sakatu

5.5.4.2 VNCren SSH bidezko ataka birbidalketa script bat erabiliz

Beste aukera bat script bat erabiltzea da, urruneko makinara SSH bidez konektatu, x11vnc zerbitzaria abiarazi eta konexioa birbidaltzeko makina lokalera. 5 segundo itxaron ondoren vinagre abiaraziko dugu makina lokalean urruneko mahaigaina ikusteko.

```
#!/bin/sh
ssh -f -L 5900:localhost:5900 asier@192.168.1.130 \
    x11vnc -safer -localhost -nopw -once -display :0 \
    && sleep 5 \
    && vinagre localhost:0
```

Script hau gorde eta egikaritzeko baimena emango diogu:

```
chmod u+x SSH_VNC_bidez.sh
```

Ondoren, egikaritu egingo dugu:

```
./SSH_VNC_bidez.sh
```

VNC zerbitzariaren pasahitza eskatuko digu eta ondoren Vinagre abiaraziko da urruneko mahaigaina erakutsiz.

Urruneko makinara ezarritako konexio bakarra SSH bidezkoa da. Egiaztatzeko netstat erabiliko dugu makina lokalean:

```
sudo netstat -tanp | grep ESTABLISHED
tcp        0      0 192.168.1.139:40526     192.168.1.130:22        ESTABLISHED 3235/ssh
tcp        0      0 127.0.0.1:34077         127.0.0.1:5900          ESTABLISHED 3246/vinagre
tcp        0      0 127.0.0.1:5900          127.0.0.1:34077         ESTABLISHED 3235/ssh
```

Berdina egingo dugu urruneko makinan:

```
sudo netstat -tanp | grep ESTABLISHED
tcp        0      0 192.168.1.130:22        192.168.1.139:40526     ESTABLISHED 1627/sshd: asier [p
tcp        0      0 127.0.0.1:47834         127.0.0.1:5900          ESTABLISHED 1695/sshd: asier@no
tcp       10      0 127.0.0.1:5900          127.0.0.1:47834         ESTABLISHED 1696/x11vnc
```

Gauza bera egingo dut orain baina oraingoan bezeroa eta zerbitzaria ez dira sare lokal berean egongo, eta konexioa interneten zehar emango da. Zerbitzariaren IPa dinamikoa denez lehen sortu dugun dyndns helbidea erabiliko dugu konexioa ezartzeko.

Egokitutako scripta ondokoa da:

```
#!/bin/sh
ssh -f -L 5900:localhost:5900 -p 2222 asier@probatzeko.dyndns.org \
    x11vnc -safer -localhost -nopw -once -display :0 \
    && sleep 5 \
    && vinagre localhost:0
```

Bezeroan egikarituko dugu:

```
./VNC_SSH_bidez_WAN_dyndns.sh
```

Vinagre urruneko mahaigainen ikustailea ireki da eta urruneko mahaigaina agertu zait, guztia behar bezala joan dela dirudi. Egiazta dezagun netstat-ekin.

Bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
tcp      0      0 127.0.0.1:5900          127.0.0.1:60047         ESTABLISHED 4147/ssh
tcp      0      0 192.168.1.139:60287     74.125.230.210:2222     ESTABLISHED 4147/ssh
tcp      0      0 127.0.0.1:60047         127.0.0.1:5900          ESTABLISHED 4149/vinagre
```

Zerbitzarian:

```
sudo netstat -tanp | grep ESTABLISHED
tcp      0      0 127.0.0.1:53553         127.0.0.1:5900          ESTABLISHED 1928/sshd: asier@no
tcp      0      0 127.0.0.1:5900          127.0.0.1:53553         ESTABLISHED 1929/x11vnc
tcp      0      0 192.168.1.130:22        192.168.1.1:60287       ESTABLISHED 1858/sshd: asier [p
```

5.5.5 VNC Firefox nabigatzailea bezero bezala erabiliz

1. Zerbitzaria martxan jarri:

```
x11vnc -forever usepw -httpdir /usr/share/vnc-java/ -httpport 5800
```

2. Bezeroan Firefox nabigatzailea ireki eta zerbitzariaren helbideko 5800 atakara konektatu:
   http://192.168.1.130:5800
3. Ondoko mezua agertuko zaigu Firefox-en :
   Orri honetako informazio guztia bistaratzeko plugin gehigarriak behar dira.
4. **Instalatu falta diren plugin-ak...** sakatu
5. The IcedTea Web Browser Plugin-a instalatzeko leihoa agertuko zaigu, egin klik **Hurrengoa**n.

6. Software gehigarria instalatu nahi dugun galdetuko digu, sakatu **Instalatu**. Aurretik instalatu gabe badugu openJDK ere instalatzen da prozesu honetan.
7. Firefox berrabiarazitakoan VNC pasahitza eskatuko digu, idatzi eta sakatu OK.
8. Urruneko ordenagailuaren mahaigaina agertuko zaigu nabigatzailearen leihoan.

Konexioa behar bezala ezarri dela egiaztatzeko netstat egikarituko dugu, lehenik bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp6 0 0 192.168.1.139:45974 192.168.1.130:5900 ESTABLISHED 6951/java

Eta ondoren, zerbitzarian:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp6 0 0 192.168.1.130:5900 192.168.1.139:45974 ESTABLISHED 3082/x11vnc

5.5.6 VNC Firefox nabigatzailea bezero bezala erabiliz SSH portu birbidalketa bidez

1. Lehen saiakeran 5900 ataka bakarrik birbidali dut, baina ez du funtzionatu. Ondoren, 5800 ataka birbidaliz saiatu naiz eta honek ere ez du funtzioantu. Azkenean bi atakak birbidali ditut:

   ```
   ssh -L 5800:localhost:5800 -L 5900:localhost:5900 asier@192.168.1.130
   ```

2. Bezeroan Firefox ireki eta ondoko helbidea idatzi:

   http://localhost:5800/

3. Pasahitza idatzi ondoren sakatu OK . Oraingoan bai, funtzionatu du.

Bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:5900 127.0.0.1:35162 ESTABLISHED 7633/ssh

tcp 0 0 192.168.1.139:42607 192.168.1.130:22 ESTABLISHED 7633/ssh

tcp6 0 0 127.0.0.1:35162 127.0.0.1:5900 ESTABLISHED 7664/java

Zerbitzarian: `sudo netstat -tanp | grep ESTABLISHED`

tcp 0 0 127.0.0.1:44749 127.0.0.1:5900 ESTABLISHED 3949/2

tcp 0 0 192.168.1.130:22 192.168.1.139:42607 ESTABLISHED 3882/sshd: asier [p

tcp 0 0 127.0.0.1:5900 127.0.0.1:44749 ESTABLISHED 3688/x11vnc

5.5.7 VNC Firefox nabigatzailea bezero bezala erabiliz SSH proxy bidez

Zerbitzaria proxy bezala erabiliz ere saiatuko naiz:

1. Bezeroan ondokoa egikaritu dut:

   ```
   ssh -ND 9999 asier@192.168.1.130
   ```

2. Bezeroko Firefox-en:

   Editatu > Hobespenak > Aurreratua > Ezarpenak

3. Eskuzko proxy-konfigurazioa hautatu.

   SOCKS ostalaria: localhost

   Ataka: 9999

4. Sakatu Ados.
5. Firefox-en http://192.168.1.130:5800/ helbidera nabigatu.

Ezarritako konexioa aztertzeko netstat erabiliko dut, lehenik bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:9999 127.0.0.1:37874 ESTABLISHED 7350/ssh

tcp 0 0 192.168.1.139:60476 192.168.1.130:22 ESTABLISHED 7350/ssh

tcp6 0 0 127.0.0.1:37874 127.0.0.1:9999 ESTABLISHED 7163/java

Eta zerbitzarian: `sudo netstat -tanp | grep ESTABLISHED`

tcp 0 0 192.168.1.130:5900 192.168.1.130:43525 ESTABLISHED 3314/x11vnc

tcp 0 0 192.168.1.130:22 192.168.1.139:60476 ESTABLISHED 3520/sshd: asier [p

tcp 0 0 192.168.1.130:43525 192.168.1.130:5900 ESTABLISHED 3587/sshd: asier

5.5.8 VNC bitxikeria: VNC zerbitzaria eta bezeroa makina berean

Zer gertatzen da VNC zerbitzaria eta bezeroa makina berean daudenean?

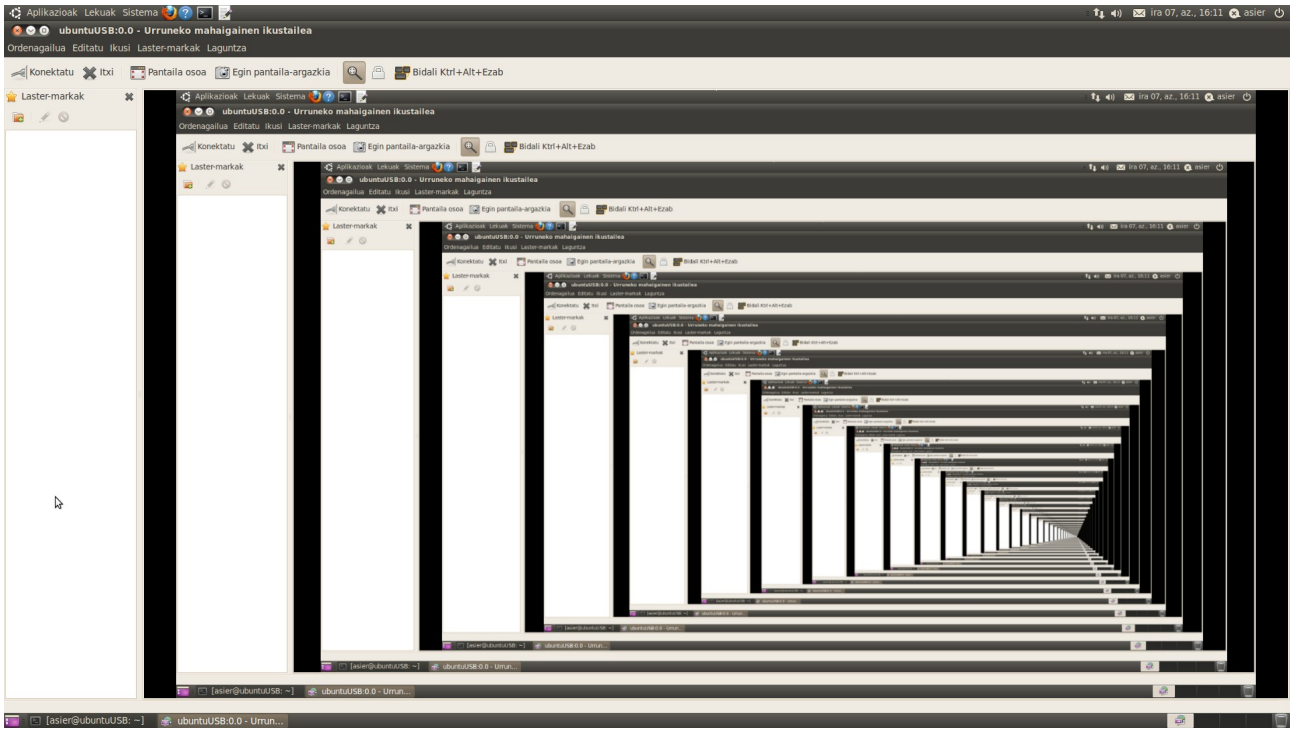
1. x11vnc zerbitzaria martxan jarri:

   ```
   x11vnc -forever -usepw -httpdir /usr/share/vnc-java/
   -httpport 5800
   ```

2. Vinagre Urruneko mahaigainen ikustailea abiarazi eta honela konfiguratu:

- **Konexioaren aukerak**
  - **Protokoloa**: VNC
  - **Ostalaria**: localhost
- **VNC aukerak**
  - **Eskalatzea**: Bai
    - **Mantendu aspektu-erlazioa**: Bai

3. Sakatu **Konektatu** eta pasahitza idatzi



4. Mahaigainaren irudi errekurtsibo bat agertu zaigu :-D

5.5.9 Bibliografia

- https://help.ubuntu.com/community/VNC
- http://kshwetabh.wordpress.com/2010/01/09/remotely-control-ubuntu-using-your-iphone-vnc-setup/
- https://help.ubuntu.com/community/VNC/Clients
- https://help.ubuntu.com/community/VNC/Servers

5.6 NX

5.6.1 FreeNX zerbitzariaren instalazioa

FreeNX instalatu aurretik SSH zerbitzaria prest dugula ziurtatu behar dugu.

1. FreeNX-ren PPA gehitu behar diogu gure sistemaren software jatorriei:
   ```
   sudo add-apt-repository ppa:freenx-team
   ```
2. Apt eguneratu:
   ```
   sudo apt-get update
   ```
3. FreeNX zerbitzaria instalatu:
   ```
   sudo apt-get install freenx
   ```
4. nxsetup scripta deskargatu behar dugu ondoren, ez baitator FreeNX PPA paketean:

```
wget https://bugs.launchpad.net/freenx-server/+bug/576359/
+attachment/1378450/+files/nxsetup.tar.gz
```

5. Deskargatu ondoren fitxategia deskonprimatuko dugu:
   ```
   tar -xvf nxsetup.tar.gz
   ```
6. Ondoren, /usr/lib/nx/ katalogora kopiatuko dugu:
   ```
   sudo cp nxsetup /usr/lib/nx/nxsetup
   ```
7. Azkenik egikaritu egingo dugu scripta:
   ```
   sudo /usr/lib/nx/nxsetup --install
   ```
8. Scripta egikaritzean gako pare berri bat sortu nahi al dugun galdetuko digu:

   Do you want to use your own custom KeyPair? [y/N]

   NoMachine-ren gakoak erabiltzea da balio lehenetsia baina ez da segurua, edozein konektatu ahal izango baita gure zerbitzarira, aurrerago ikusiko dugu gako pertsonalizatuak nola erabili.

FreeNX zerbitzaria abiarazteko:

```
sudo service freenx-server start
```

geldiarazteko:

```
sudo service freenx-server stop
```

eta egoera jakiteko:

```
status freenx-server
```

5.6.2 Neatx zerbitzariaren instalazioa

1. Lehenik freenx-team-en PPA gehituko behar da gure software jatorrietara. Lehen bi pausoak

ez dira beharrezkoak nire kasuan, lehen gehitu baitut PPA hau freenx-server instalatzeko.

```
sudo add-apt-repository ppa:freenx-team
```

2. Ondoren, gure software jatorriak eguneratuko ditugu:

```
sudo apt-get update
```

3. apt-get bidez instalatu:

```
sudo apt-get install neatx-server
```

5.6.3 QTNX bezeroa instalatu eta probak egin

Bezero bezala QTNX erabiliko dugu, horretarako:

```
sudo apt-get install qtnx
```

Bezeroa abiarazteko **Aplikazioak > Internet > QTNX**

5.6.3.1 FreeNX zerbitzariarekin proba

Lehenik, FreeNX zerbitzariarekin egingo dugu proba.

Erabiltzaile izena, pasahitza eta konfigurazioko zenbait aukera zehaztu beharko dira ondoren, ostalariaren helbidea, sistema eragilea, mahaigain mota, erresoluzioa, ...

- **Username**: asier
- **Password**: **********
- **Session**: Create new session LAN

Eta **Configure...** sakatu, **Basic** fitxan ondoko datuak sartu:

- **General**
  - **Session Name**: ProbatzenNX (Gero aurreko menuko Session-en agertuko den erreferentzia)
- **Server**
  - **Hostname**: 192.168.1.100
  - **Port**: 22
  - **Use default key**: Bai
- **Desktop**
  - **Platform**: UNIX (Beste aukerak: WINDOWS, VNC Proxy)
  - **Type**: GNOME (Beste aukerak: KDE, CDE, XDM, Custom)
  - **Link**: LAN (Beste aukerak: Modem, ISDN, ADSL, WAN)
- **Geometry**
  - **Resolution**: 640x480 (Beste aukerak: 800x600, 1024x768, Full Screen, Custom)
  - **Compression**: PNG (Beste aukerak: JPEG, Raw X11)
  - **JPEG Quality**: 5/8

    - **Use RENDER Extension**: Bai

Eta **Advanced** fitxan berriz ondokoak:

- **Network**
    - **Use SSH Tunneling**: Bai
- **Cache**
    - **Memory:** 8 MB
    - **Disk**: 32 MB

Ondoren **Apply** sakatu konfigurazioa aplikatzeko eta ondoren **Connect** konektatzeko.

QTNX erabiltzen dugun hurrengo aldian konfigurazio hau eskuragarri izango dugu Session Name bezala jarri diogun izenarekin (ProbatzenNX) Session-en, honi esker konfigurazioa behin bakarrik egin beharko dugu.

Makina lokalean:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:59376 192.168.1.130:22 ESTABLISHED 3375/ssh

Urruneko makinan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:60203 127.0.0.1:6000 ESTABLISHED 2818/netcat
tcp 0 0 127.0.0.1:22 127.0.0.1:52734 ESTABLISHED 2191/sshd: asier [p
tcp 0 0 127.0.0.1:6000 127.0.0.1:60203 ESTABLISHED 2785/nxagent
tcp 0 0 127.0.0.1:52734 127.0.0.1:22 ESTABLISHED 2186/ssh
tcp 0 0 192.168.1.130:22 192.168.1.139:59376 ESTABLISHED 1959/sshd: nx [priv

Konexioa sare lokalean ezarri ordez interneten zehar ezarri nahi bagenu, ondoko aukerak hautatu beharko genituzke, gainerakoak lehen bezala ondo daude:

- **Hostname**: adibidea.dyndns.org
- **Port**: 2222
- **Link**: WAN

Eta **Connect** sakatu ondoren urruneko mahaigaina agertuko zaigu, arazorik gabe.

Egiazta dezagun konexioa netstat-ekin, lehenik makina lokalean:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:42563 74.125.230.210:2222 ESTABLISHED 4195/ssh

Eta ondoren, zerbitzarian:

```
netstat -tanp | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:6000 127.0.0.1:43033 ESTABLISHED 2869/nxagent

tcp 0 0 192.168.1.130:22 192.168.1.1:42563 ESTABLISHED 2038/sshd: nx [priv

tcp 0 0 127.0.0.1:45428 127.0.0.1:22 ESTABLISHED 2265/ssh

tcp 0 0 127.0.0.1:43033 127.0.0.1:6000 ESTABLISHED 2887/netcat

tcp 0 0 127.0.0.1:22 127.0.0.1:45428 ESTABLISHED 2270/sshd: asier [p

Bi kasuetan ikusten denez, urruneko makinara ezarritako konexio bakarra SSH bidezkoa da.

Ondorio hauek atera nituen:

- QTNX bezeroarekin konektatzean erabiltzaile jakin bat bezala saio bat hasten badugu, adibidez, asier bezala eta urruneko makinan dagoeneko saio bat hasita badago ere erabiltzaile beraren izenean bi saioak ez dira berdinak. Hau da, ez da VNC bezala, non mahaigain bera partekatzen duten zuzenean makinara konektatutako erabiltzaileak eta urrunetik VNC bezero bidez konektatutakoak.
- Makina berean saio bat baino gehiago egon daitezke aldi berean, adibidez, makina batean "asier" erabiltzaileak saio bat hasita badauka ere urruneko erabiltzaileak NX bidezko beste saio bat has dezake "mikel" erabiltzailearen kontua erabiliz.
- Urruneko NX saio batean ordenagailua itzaltzen saiatzen bagara supererabiltzaile pasahitza eskatuko zaigu.
- VNC baino azkarragoa da, ez dirudi urruneko makina bat denik.

5.6.3.2 Neatx zerbitzariarekin proba

NX zerbitzarian FreeNX-server geldituta dagoela egiaztatuko dut:

```
status freenx-server
```

freenx-server stop/waiting

Ondoren, QTNX bezeroa erabiliz konektatu naiz, FreeNX zerbitzarira konektatzeko erabilitako aukera berdinak erabili ditut eta funtzionatzen du.

Egiazta dezagun konexioa behar bezala ezarri den, lehenik bezeroan:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.139:46659 192.168.1.130:22 ESTABLISHED 2056/ssh

Eta konexioa egiaztatuko dut netstat-ekin:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 192.168.1.130:22 192.168.1.139:46659 ESTABLISHED 2360/sshd: nx [priv
tcp 0 0 127.0.0.1:34566 127.0.0.1:22 ESTABLISHED 2587/ssh
tcp 0 0 127.0.0.1:44038 127.0.0.1:6000 ESTABLISHED 3208/netcat
tcp 0 0 127.0.0.1:22 127.0.0.1:34566 ESTABLISHED 2592/sshd: asier [p
tcp 0 0 127.0.0.1:6000 127.0.0.1:44038 ESTABLISHED 3193/nxagent

Ez dut aurkitu nola gelditu eta abiarazten den neatx zerbitzaria.

Neatx-en konfigurazio adibidea: /etc/neatx.conf

5.6.4 OpenNX bezeroaren instalazioa eta probak

Lehenik instalatu egingo dugu:

1. Gako hau deskargatu:
   http://download.opensuse.org/repositories/home:/felfert/Debian_5.0/Release.key
2. Gakoa inportatu
   ```
   sudo apt-key add /home/asier/Deskargak/Release.key
   ```
   OK
3. /etc/apt/sources.list fitxategiari ondoko lerroa gehitu:
   deb http://download.opensuse.org/repositories/home:/felfert/xUbuntu_10.04 ./
4. Software jatorriak eguneratu:
   ```
   sudo apt-get update
   ```
5. Apt-get bidez instalatu:
   ```
   sudo apt-get install opennx
   ```

Ondoren, proba batzuk egingo ditugu.

Hiru aplikazio berri agertu zaizkit menu nagusiko **Aplikazioak > Internet** atalean:

OpenNX client

OpenNX Connection Wizard

OpenNX Session Administrator

1. **Aplikazioak > Internet > OpenNX client**
2. **OpenNX Konexio Laguntzailea** (OpenNX Connection Wizard) agertu da, **Next** sakatu

3. Saioaren datuak sartu:
   - **Session**: OpenNX_FreeNX

- **Host**: 192.168.1.130
- **Port**: 22
- **Select the type of your internet connection**: LAN

Eta **Next** sakatu.

5. Erabili nahi dugun **zerbitzua** aukeratu: Unix(NX), VNC, Windows(RDP), Shadow (?). Unix hautatuko dugu.

Mahaigain mota: GNOME

Eta **urruneko mahaigainaren tamaina**: Available Area (Beste aukerak: 640x480, 800x600, 1024x768, Full Screen eta Custom) .

Ondoren sakatu **Next**.

5. **Enable SSL encryption of all traffic**: Bai

   Berriz ere **Next**.
6. Konexioa konfiguratu dugu. Gure makinaren mahaigainean lasterbide bat sortzeko aukera emango zaigu eta konfigurazio aurreratuarekin jarraitzeko aukera.
7. Berriz ere, **Aplikazioak > Internet > OpenNX Client**
8. Saioa abiarazteko leihoa agertu zait, bertan datu hauek sartu behar dira:
   - **Login**: asier
   - **Password**: *********
   - **Session**: OpenNX_FreeNX (Oraintxe sortu dugun konfigurazioa)

   Ondoko aukerak eskaintzen zaizkigu:
   - **Use SmartCard** (SmartCard-a erabili): Ez
   - **Login as guest user** (Saioa gonbidatu bezala hasi): Ez
   - **Configure...** botoia sakatuz konexioa konfigura dezakegu berriz ere.
9. **Login** sakatu

Errorea eman dit:

execvp(/usr/bin/nxssh, -nx, -x, -2, -p, 22, -o, RhostsAuthentication no, -o, PasswordAuthentication no, -o, RSAAuthentication no, -o, RhostsRSAAuthentication no, -o, PubkeyAuthentication yes, -i, /usr/share/keys/server.id_dsa.key, -E, nx@192.168.1.130) failed with error 2!

Aplikazioaren sourceforge-ko foroan aurkitu dut soluzioa:

http://sourceforge.net/tracker/?func=detail&aid=2829183&group_id=184662&atid=910152

Eskaintzen duen lehen soluzioa ez dut ondo ulertu, OpenNX-en ingurune konfigurazio fitxategia (**~/.opennx**) ezabatu, opennx **/usr/NX/bin/** katalogoan instalatu eta berriro saiatu behar omen da.

Eskaintzen duen bigarren aukera "instalatu gabe aplikazioa probatzeko" omen da:

1. **Configure...** hautatu
2. System NX dir zuzendu behar da, **/usr/** agertzen da eta **/usr/NX/** izan behar du.
3. **Apply** eta **OK** sakatu.
4. Berriz saiatu konektatzen.

Oraingoan ondo konektatu da urruneko mahaigainera eta behar bezala funtzionatzen du.

Proba hau egin ondoren, hurrengo egunean ordenagailua berrabiarazi nuenean OpenNX menu nagusitik desagertuta zegoen. Behar bada “instalatu gabe aplikazioa probatzeko” aukera hautatu nuelako. Aurkitu nuen soluzioan aipatzen zen lehen aukera jarraitu beharko nuen behar bada.

5.6.5 Bibliografia

- https://help.ubuntu.com/community/FreeNX
- http://www.ubuntugeek.com/how-to-install-neatx-similar-to-freenx-server-on-ubuntu-10-04-lucid-lynx.html
- Freenx/neatx konfigurazio gida:

  http://virtulinux.blogspot.com/2011/01/configurar-servidor-terminal-server-en.html
- OpenNX instalazio gida:

  http://opennx.net/download.html

5.7 Androiderako aplikazioak garatzeko ingurunea prestatu

5.7.1 Android SDK eta ADT-ren instalazioa

Lehenik eta behin Android SDK instalatuko dugu:

1. *SDK Starter Package* deskargatu. http://developer.android.com/sdk/index.html
2. tgz fitxategia deskonprimitu eta gogoan hartu non jartzen dugun, gero beharko baitugu. Nik /home/asier/ katalogoan deskonprimitu dut.

Ondoren Eclipserako **Android Development Tool** (ADT) plugin-arekin jarraituko dugu:

ADT plugina instalatu ahal izateko beharrezkoak dira ondokoak:

- Eclipse 3.5 (Galileo) edo berriagoa
- Eclipseren JDT plugina
- JDK 5 edo JDK 6 (JRE ez da nahikoa)

1. Eclipse abiarazi eta **Help > "Install New Software..."** menuan klikatu
2. **Add** botoia sakatu
3. **Add Repository** elkarrizketan idatzi "ADT Plugin" izena eta https://dl-ssl.google.com/android/eclipse/ URL-a. Arazorik izanez gero erabili http https-ren ordez.
4. Klikatu **OK**.
5. **Available Software** elkarrizketan hautatu **Developer Tools** eta sakatu **Next**.
6. Hurrengo leihoan deskargatuko diren tresnen zerrenda agertuko da, sakatu **Next**.
7. Irakurri eta onartu lizentzia eta sakatu **Finish**.
8. Bukatzean Eclipse berrabiarazi.

5.7.2 ADT pluginaren konfigurazioa

Aurreko atalean ADT plugina instalatu ondoren beharrezkoa da Android SDK-ren kokapena zehaztea:

1. Klikatu **Window > Preferences...** hobespenen leihoa irekitzeko.
2. Hautatu **Android** ezkerreko panelean.
3. Elkarrizketa bat agertuko da erabilera estatistikak Google-ri bidali nahi dizkiogun galdetuz. Aukeratu bai edo ez eta sakatu **Proceed**.
4. Klikatu **Browse** eta hautatu SDK kokapena.
5. Sakatu **Apply** eta ondoren **OK**.

5.7.3 Android SDK-ri osagaiak gehitzea

1. Ireki **Android SDK and AVD Manager**
2. **Available packages** menua klikatu ezkerreko menuan

3. **Android Repository** aukeratu eta bertatik instalatu nahi ditugun osagaiak hautatu. Gomendatuak **Android SDK Platform-tools** eta gutxienez plataforma bat dira (adibidez **SDK Platform Android 1.6, API 4, revision 3** edo **SDK Platform Android 2.1, API 7, revision 3**). Dokumentazioa eta instalatu ditugun plataformari dagozkien adibideak (samples) instalatzea ere egokia izan daiteke.
4. **Install selected** klikatu eta instalazioa hasiko da

5.7.4 Virtual Device bat sortzea

1. **Android SDK and AVD Manager** abiarazi behar da lehenik.
2. Ondoren ezkerreko zerrendan **Virtual devices** hautatu ondoren, sakatu **New...** botoia.
3. **Create new Android Virtual Device (AVD)** leihoa agertuko zaigu. Bertan hainbat aukera zehaztu beharko ditugu, makinaren izena, erabili nahi dugun Androiden bertsioa, SD txartelaren tamaina, hardwarearen ezaugarriak eta abar.
4. Ondoren **Create AVD** sakatu behar da makina birtuala sortzeko.
5. Orain **Virtual Devices** zerrendan agertuko zaigu sortu berri dugun makina birtuala, martxan jartzeko nahikoa da hautatu eta **Start...** sakatzea.

5.7.5 Android Market-a emulatzailean instalatzea

5.7.5.1 Android 1.6

Probak egiteko Android SDK and AVD Manager-ekin sortutako Android 1.6 emulatzaile bat erabiliko dut.

Android 1.cd6 sistema irudi bat deskargatu behar da lehenik. Adibidez, hemendik: http://www.4shared.com/file/165624746/fc72c3ed/system.html

HTC-ren webgunetik ere deskarga daiteke (http://developer.htc.com/adp.html).

Ondoren, sortu dugun emulatzailearen irudia gordetzen dagoen katalogoa aurkitu behar dugu, **<android-sdk-katalogoa>/platforms/android-4/images/**-en egon ohi da. Android 1.6 izan ordez beste bertsio bat izango balitz 4-ren ordez dagoen API bertsioa daukan katalogoa aukeratu beharko genuke.

Adibidez, hau da nire **<android-sdk-katalogoa>/platforms/** katalogoaren edukia:

drwxr-xr-x 10 asier asier 4096 2011-06-18 17:09 ./

drwxrwxr-x 12 asier asier 4096 2011-06-22 00:30 ../

drwxr-xr-x 6 asier asier 4096 2011-06-18 15:24 android-10/

drwxr-xr-x 7 asier asier 4096 2011-06-18 17:09 android-11/

drwxr-xr-x 7 asier asier 4096 2011-06-18 16:46 android-12/

drwxr-xr-x 9 asier asier 4096 2011-01-03 15:31 android-3/

drwxr-xr-x 9 asier asier 4096 2011-01-03 15:15 android-4/

drwxr-xr-x 8 asier asier 4096 2011-01-03 14:54 android-7/

drwxr-xr-x 8 asier asier 4096 2011-01-03 14:32 android-8/

drwxr-xr-x 6 asier asier 4096 2011-01-03 14:12 android-9/

Bertako **android-4/images/** katalogora sartuko gara ondoren eta badaezpada ere **system.img** fitxategiaren segurtasun kopia bat egingo dugu:

```
cp system.img system.img.old
```

Deskargatu dugun irudiarekin ordezkatuko dugu **system.img**:

```
cp /home/asier/Deskargak/system.img system.img
```

Android SDK and AVD Manager abiaraziko dugu eta bertatik gure emulatzailea martxan jarriko dugu. Behin emulatzailea martxan jarrita, hurrengo pausoak:

- Androidean klik egin.
- Tutoriala ikusi nahi izanez gero sakatu **Begin**, bestela **Skip**.
- Saioa hasi Google kontuarekin aukeratuz gero teklatua irekitzeko esango digu, **Skip** sakatuko dugu eta geroago hasiko dugu saioa.
- **Google Location**. Nik bi aukerei ezetz esan diet.
- Aplikazioen menuan **Market** agertuko zaigu orain, egin klik.
- Google kontu batekin lotzeko esango digu berriz ere, sakatu **Next**.
- Saioa hasteko **Sign In** sakatu, kontu bat sortzeko **Create**.
- Erabiltzaile izena eta pasahitza sartu eta sakatu **Sign In**.
- **Android Market Terms of Service**-ak onartu **Accept** sakatuz.
- Eta kitto, lortu dugu market-a emulatzailean martxan jartzea.

5.7.5.2 Android 2.2

Android 1.6-ren emulatzailean Market-a instalatzeko jarraitu beharreko pauso berdinak jarraitu behar dira baina irudi hau erabiliz: http://www.4shared.com/get/qjANIsHH/system.html

Irudia **<android-sdk-katalogoa>/platforms/android-8/images/** katalogora kopiatu behar da kasu honetan.

Oharrak:

- Emulatzailea sortzerakoan 96 MBeko cache partizioa zehaztu behar da.
- Dirudienez Android Market aplikazioa ez da eguneratzen gailuaren erabiltzaile interfazera egokitzeko.
- Aplikazio kopuru mugatu bat agertzen da Market-ean.

5.7.6 Bibliografia

- http://developer.android.com/sdk/installing.html
- Nola lortu Android Market erabili ahal izatea Android 1.6 emulatzailean:
  http://makingmoneywithandroid.com/2011/04/how-to-run-android-market-on-the-emulator/
- Nola lortu Android Market erabili ahal izatea Android 2.2 emulatzailean:
  http://techdroid.kbeanie.com/2011/01/android-market-on-emulator-22.html

5.8 Androiderako aplikazioak programatzearen oinarriak ikasten hasi

Androiderako aplikazioak nola programatu ikasten hasi naiz ondoko liburuak erabiliz:

- Learning Android, Marko Gargenta, O'Really
- Professional Android 2 Application Development, Reto Meier, Wrox

5.9 Android-erako aplikazioen itzulpena

5.9.1 Euskalbar-en instalazioa

1. https://addons.mozilla.org/eu/firefox/addon/euskalbar/ web orrira nabigatu eta + **Gehitu Firefox(e)ra** botoia sakatu
2. Agertuko den leihoan **Instalatu orain** sakatu
3. Firefox berrabiarazi

5.9.2 ConnectBot-en itzulpena

ConnectBot aplikazioaren itzulpenak Launchpad erabiliz egiten dira.

https://translations.launchpad.net/connectbot/trunk/+pots/fortune/eu/+translate

ConnectBot-en itzulpenaz Launchpad Translators taldea arduratzen da eta euskarazko itzulpenaz Launchpad Basque Translators. Itzulpenak finkatzeko baimenik ez nuenez lehenik itzulpen iradokizunak egin nituen ConnectBot-entzat. Ondoren, Launchpad Basque Translators taldearekin bat egitea eskatu nuen, eta aurretik itzulpentxo batzuk eginak nituenez launchpad-ean laster batean onartu ninduten. Ondoren, egindako iradokizunak gainbegiratu eta finkatu nituen.

Nire orria Launchpad-en:

https://launchpad.net/~asier-iturralde

Launchpad Basque Translators:

https://launchpad.net/~lp-l10n-eu

5.9.3 Gainerako aplikazioen itzulpenak: SSH-Tunnel, android-vnc-viewer eta android wake-on-lan

3 aplikazio hauen kasuan res/values/strings.xml fitxategiak eskuratu nituen bakoitzaren kode biltegitik eta gedit erabiliz itzuli nituen.

5.9.3.1 Android-vnc-viewer

Abuztuaren 30ean proiektuko arduradun bati bidali nion itzulitako xml fitxategia e-posta bidez (herbage@gmail.com). Baina oraindik ez dut erantzunik jaso.

Irailaren 8an beste arduradun bati bidali diot mezua (googlecode@antlersoft.com), aldaketen arduraduna dela dirudi. Erantzuten ez badu, code.google.com-era fitxategia igotzen saiatu beharko naiz.

5.9.3.2 SSH-Tunnel

Abuztuaren 30ean proiektuko arduradun bati bidali nion itzulitako xml fitxategia e-posta bidez (lch@fudan.edu.cn).

Abuztuaren 31n erantzun zidan, proiektuaren kode biltegian integratu zuela esanez:

http://code.google.com/p/sshtunnel/source/detail?r=f045e578e75e63fd3e95886f65731bb9692c8e45

5.9.3.3 Android wake-on-lan

Ez dut kontakturako e-postarik aurkitu. Github-en dago proiektua, erregistratu eta bertara igo beharko dut fitxategia.

5.9.4 Bibliografia

- https://help.launchpad.net/Translations
- https://help.launchpad.net/Translations/GuidesList/Basque
- https://help.launchpad.net/Translations/GuidesList/Basque/hiztegia

5.10 Aztertutako tekniketarako Androiderako aplikazio libreak probatu

5.10.1 ConnectBot

1. Google Market-ean ConnectBot aplikazioa aurkitu eta **Install** sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
3. Aplikazioen menuan **ConnectBot**-en ikonoan klik egin abiarazteko.
4. Apache 2.0 lizentzia duela jakinaraziko zaigu, **Agree** sakatu.
5. Ohar eta gomendio batzuen ondoren konektatu nahi dugun makinako erabiltzaile-izena eta ostalariaren IPa edo izena idazteko testu kutxa agertuko zaigu, erabili nahi dugun konexio mota ere aukeratu ahal izango dugu (**SSH**, telnet edo lokala). Bertan **asier@192.168.56.2** idatziko dugu eta **Done** sakatu.
6. Konektatzen garen lehen aldia denez ondoko mezua agertuko zaigu:

   Connecting to 192.168.56.2:22 via ssh

   The authenticity of host '192.168.56.2' can't be established.

   Host RSA key fingerprint is ...

   Are you sure you want to continue connecting?

   Baietz erantzungo dugu.
7. Ondokoa azalduko zaigu ondoren:

   Using algorithm: aes256-ctr hmac-sha1-96

   Trying to authenticate

   Attempting 'password' authentication
8. Agertuko zaigun testu kutxan urruneko makinako erabiltzailearen pasahitza idatzi eta saioa hasiko dugu.
9. Behar bezala dabilela egiaztatzeko fitxategi bat sortuko dut:

   ```
   touch kaixo.txt
   nano kaixo.txt
   ```

   Nahi nuen testua idatzi ondoren Ctrl botoia erabili behar da gordetzeko (Ctrl+O) eta nano ixteko (Ctrl+X), horretarako pantailan klik egitean **CTRL** botoi bat agertzen den botoia erabili behar da.
10. Urruneko makina bezala erabiltzen ari naizen makina birtualean egiaztatu dut fitxategia benetan han dagoela eta idatzi berri dudan testua han dagoela.

    ```
    ls | grep kaixo
    ```

    kaixo.txt

    ```
    cat kaixo.txt
    ```

    kaixo asier
11. Konexioa amaitzeko:

    ```
    exit
    ```

Egikaritzean ondoko mezua agertuko da:

logout

Connection Lost

Eta ConnectBot-ek ondoko galdera egingo digu:

Host has disconnected. Close session?

Baietz erantzungo dugu.

12. Momentu honetatik aurrera gure urruneko makina **ConnectBot:Hosts** zerrendan agertuko da eta birkonektatzeko ez dugu erabiltzaile-izen eta ostalariaren IP edo izena idatzi beharrik izango, nahikoa izango da klik egitea.

Benetako Android gailu batekin ere egin dut proba interneten zehar adibidea.dyndns.org-ekin.

5.10.2 VNC birbidalketa ConnectBot erabiliz

1. **ConnectBot** abiarazi.
2. SSH konexioa ezarri urruneko VNC zerbitzarira:

   ```
   asier@192.168.1.130
   ```

   Pasahitza eskatuko digu.
3. **MENU** botoia sakatu eta **Port Forwards** klikatu
4. Ataka birbidalketa bat egiten dugun lehen aldia denez ondoko mezua agertuko zaigu:

   Tap Menu to create port forwards
5. **MENU** sakatuko dugu eta **Add port forward** klikatu
6. Ondok datuak bete behar ditugu:

   Nickname: VNC birbidalketa

   Type: Local (Beste aukerak: Remote, Dynamic (SOCKS))

   Source port: 5900

   Destination: localhost:5900
7. Eta **Create port forward** klikatu
8. Sortu berri dugun ataka birbidalketa 4. puntuko zerrendan agertuko zaigu.
9. VNC zerbitzaria abiaraziko dugu **ConnectBot** erabiliz, ondoko komandoa egikarituz:

   ```
   x11vnc -safer -localhost -nopw -once -display :0
   ```
10. **Android VNC viewer** abiarazi. Aukera hauekin:
    - **Connection**: New
    - **Nickname**: VNC birbidalketa
    - **Password**: ********* (pasahitza) (ez da beharrezkoa, -nopw erabili baitugu)
    - **Address**: localhost
    - **Port**: 5900

11. **Connect** sakatu eta VNC zerbitzariak ssh bidez bidalitako mahaigaina agertuko zaigu gure gailuan.
12. Egiaztatzeko netstat erabiliko dugu zerbitzarian:

```
sudo netstat -tanp | grep ESTABLISHED
```

tcp 0 0 127.0.0.1:5900 127.0.0.1:55043 ESTABLISHED 1729/x11vnc

tcp 0 0 192.168.1.130:22 192.168.1.139:59699 ESTABLISHED 1599/sshd: asier [p

tcp 0 0 127.0.0.1:55043 127.0.0.1:5900 ESTABLISHED 1667/0

13. VNC zerbitzaria gelditu eta konexioa amaitzeko, ConnectBot irekiko dugu berriz ere:
    - VNC zerbitzaria gelditzeko: Ctrl + c
    - SSH konexia eteteko: exit

Benetako Android gailu batekin ere egin dut proba interneten zehar adibidea.dyndns.org-ekin.

5.10.3 SSH Tunnel

SSH tunel bat sortzeko SSH zerbitzari bat instalatuta duen makina bat beharko dugu eskuragarri. Ondoren, gure Android gailuan ondoko pausoak jarraitu beharko ditugu:

1. Google Market-ean SSH Tunnel aplikazioa aurkitu eta **Install** sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
3. Aplikazioen menuan **SSHTunnel** ikonoan klik egin abiarazteko.
4. Ondoko aukerak eskaintzen dizkigu aplikazioak:
    - **Service Controller**
        - **Tunnel Switch**: Bai (Bai hautatzean SSH tunela gaitzen da eta ez hautatzean desgaitu)
        - **Profiles**: Default Profile (Ezarpenak gordetzek aukera eskaintzen digu, aukerak profil batean bilduz. Profil bat ezartzeko nahikoa izango da Profiles-en klik egin eta hautatzea)
    - **SSH Tunnel Settings**
        - **Host**: 192.168.0.101 (Konektatu nahi dugun zerbitzariaren IP helbidea)
        - **Port**: 22 (Konexioan erabili nahi dugun ataka)
    - **Account Information**
        - **User**: asier (Zerbitzariko erabiltzaile-izena)
        - **Password/Passphrase**: ********** (Hautatutako erabiltzaileari dagokion pasahitza edo pasaesaldia)
    - **Port Forwarding**
        - **Use socks proxy**: Bai (Bai hautatuz gero, ataka birbidalketa dinamikoa aktibatzen da, ataka guztiak birbidaliz)
        - **Local port**: 1984

- **Remote address**: 127.0.0.1
- **Remote port**: 3128

- **Feature Settings**
  - **Autoconnect**: Ez (Bai hautatuz gero, sarerako konexioa dagoen bezain pronto aktibatuko da SSH tunela)
  - **Binded Network**:(Proxy ezarpenak 2G/3G sare bati edo zehazturiko SSID-ari lotu)
  - **Auto Reconnect**: Bai (SSH zerbitzura birkonektatzen saiatu ustekabean konexioa eteten denean)
  - **Enable GFW List**: Ez (Blokeatutako guneentzat bakarrik ezarri proxy-a, Txinako erabiltzaileentzat soilik)
  - **Global Proxy**: Ez (Proxy globala ezarri automatikoki)
  - **Individual Proxy**: Klik egiten badugu gailuan instalatuta ditugun aplikazioen zerrenda bat agertuko zaigu, bertatik proxy-arekin erabili nahi ditugunak hautatzeko. Adibidez, Browser hautatu dut, modu pribatuan nabigatu ahal izateko.
  - **Enable DNS Proxy**: Bai (Gaitu DNS Proxy-a suhesiaren atzeko zenbait erabiltzailerentzat)
- **Notification settings**
  - **Show notification icon**: Bai (Jakinerazpen bat dagoenean ikono bat agertzea nahi dugun hautatzeko aukera eskaintzen digu)
  - **Select ringtone**: (Jakinerazpen bat dagoenean erreproduzituko den soinua hautatzeko aukera eskaintzen digu)
  - **Vibrate**: Bai (Jakinerazpen bat dagoenean edo konexioaren egoera aldatzean bibratzerik nahi dugun hautatzeko aukera eskaintzen digu)

5. Ezarpenak konfiguratu ondoren **Tunnel Switch** klikatuko dugu SSH Tunela aktibatzeko.
6. Pantailan **Connecting...** agertuko zaigu eta gauzak ondo badoaz **Connected** eta pantailaren goiko aldean SSHTunnel aplikazioaren ikonoa agertuko zaigu, planeta bat eraztun batekin.
7. Nabigatzailea ireki eta modu pribatuan nabigatu.

Oharra: Gako pribatu/publikoak erabili nahi izanez gero /sdcard/sshtunnel/key fitxategian gorde.

SSH tunelak erabiltzeko beste aukera bat ConnectBot erabiltzea da.

5.10.4 Wake-On-Lan

1. Google Market-ean Wake On LAN (Egilea: Mafro) aplikazioa aurkitu eta **Install** sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
3. Urruneko makina itzalita baina pakete magikoak jasotzeko prestatuta dagoela ziurtatu ondoren, android gailuan aplikazioen menuan **Wake on LAN** ikonoan klik egingo dugu abiarazteko.
4. Bi fitxa agertuko zaizkigu: History eta Wake. **Wake** klikatuko dugu.

5. Bertan ondoko testu kutxak agertuko zaizkigu:
   - **Reference**: Makina identifikatzeko erabil dezakegu
   - **MAC Address**: Altxa nahi dugun makinaren MAC helbidea
   - **Host Name / IP Address**: Altxa nahi dugun makinaren IP helbidea
   - **Port**: Erabiliko dugun ataka. Aplikazioak lehenesten duena: 9

Ondoko datuekin bete ditut:

- **Reference**: Probatzen1
- **MAC Address**: A1:B2:C3:D4:E5:F6
- **Host Name / IP Address**: 192.168.0.101
- **Port**: 7

Ondoren, **Send Wake Packet** sakatu dut baina makina ez da piztu.

Android 1.6 emulatzailean Wake-on-LAN aplikazioa erabiliz ez dut lortu ordenagailua pizterik, ez sare lokalean (paketea bidaltzen du baina ez da pizten), ez interneten zehar (adibidea.dyndns.org helbidea erabiliz).

Android 1.6 duen gailu erreal batekin ere egin dut aproba eta kasu honetan bai, lortu dut ordenagailua piztea. Baina ez du beti ondo funtzionatzen. Zer gertatzen den aztertu beharko litzateke.

5.10.5 Android-vnc-viewer

1. Google Market-ean android-vnc-viewer aplikazioa aurkitu eta **Install** sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
3. Urruneko makinan vnc zerbitzaria martxan dagoela ziurtatu ondoren, android gailuan aplikazioen menuan **VNCviewer** ikonoan klik egingo dugu vnc bezeroa abiarazteko.
4. Agertuko zaigun menuan hainbat aukera daude:

   Connection: Konexio konfigurazioen zerrenda. Konexio konfigurazio berri bat prestatzeko New hautatu behar da.

   Nickname: Sortzen dugun konfigurazio bakoitzak ezizen bat eduki dezake, zerrendan bilatzea errazteko. Ez du zerikusirik erabiltzaile-izenarekin.

   Password: VNC zerbitzaria pasahitza eskatzeko konfiguratuta badago, testu kutxa honetan idatzi behar da pasahitz hori. Keep hautatuz gero pasahitza gailuko datubasean gordeko da bestela konektatzen garen aldi bakoitzean idatzi beharko dugu. Pasahitza datu basean ez gordetzea hautatu arren aplikazioa martxan dagoen bitartean memorian gordeko da.

   Address: VNC zerbitzaria dagoen makinaren DNS izena edo IP helbidea.

   Port: VNC zerbitzariak erabiltzen duen ataka. Balio lehenetsia 5900 da.

   Color Format: Bezeroak hainbat kolore formatu desberdin erabili ditzazke, pixeleko kolore kopuru/bit kopuru bezala adierazten dira. Kalitate handiagoko kolore formatuak ikusgarriagoak dira baina banda zabalera handiagoa behar dute eta Android gailuaren CPU-

aren erabilera handitzen dute. VNC zerbitzari guztiek ez dituzte kolore formatu guztiak onartzen, adibidez, OS/X Remote Desktop-ek 24-biteko kolore formatua erabiltzera behartzen du. Arazoak izanez gero, saiatu beste formatu batekin.

Use local mouse pointer: Zenbait VNC zerbitzarik (OS/X-erakoak adibidez) ez dute saguaren erakuslerik marrazten bezeroan, honek erabilera asko zailtzen du, ezinezko bihurtzen baitu erakuslea non dagoen jakitea. Aukera hau hautatuz gero android-vnc-viewerrek erakusle bat marraztuko du saguaren posizioan.

Force full-screen bitmap: Mahaigain handi baten irudia zerbitzatzen ari den VNC zerbitzari batek bidalitako bitmap irudia Androidek aplikazioari onartutako memoria kopurua baino handiagoa izan daiteke. Mota honetako mahaiganekin lan egiteko, android-vnc-clientek irudia hainbat lauzatan zatitzen du. Honek zoritxarrez, kasu batzuetan, ikuskapen arazoak sortzen ditu. Force full-screen bitmap hautatuz gero beti mahaigain osoa agertuko da eta ez da zatiketarik emango, baina honek aplikazioa kraskatzea eragin dezake.

Repeater: Aukera hau hautatuko dugu VNC zerbitzarira konektatzeko UltraVNC errepikagailu bat erabiltzen ari bagara. Kasu horretan, errepikagailuaren VNC zerbitzari helbidea eta pantaila id-a edo ataka zenbakia idatzi beharko ditugu dagokion testu kutxan bi puntuz banaturik.

Ondokoak datuak zehaztuko ditugu:

Connection: New

Nickname: asier

Password: **********

Address: 192.168.56.2

Port: 5900

Gainerako aukerak dauden bezala utziko ditugu.

5. **Connect** sakatu
6. Android gailuaren pantailan agertuko zaigu urruneko mahaigaina. Pixka bat mantsoa bada ere erabilgarria da, programak abiarazi eta abar egin daiteke eta ondo erantzuten du.
7. Deskonektatzeko sakatu android gailuaren **MENU** botoia eta ondoren pantailan agertuko diren botoien artean **Disconnect** sakatu.
8. Konektatzen saiatzen garen hurrengo aldian ez dugu datu guztiak sartu beharrik izango. Nahikoa izango da Connection-en asier:192.168.56.2:5900 hautatzea eta Connect sakatzea. Makina berri batera konexio bat sortu nahi bagenu berriz lehen bezala Connection: New hautatu beharko genuke.

5.10.6 droid-vnc-server

1. Google Market-ean droid VNC server BETA aplikazioa aurkitu eta Install sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
3. Aplikazioen menuan **VNC server BETA** ikonoan klik egingo dugu abiarazteko.
4. Ezin duela aurrera jarraitu esango digu, aurretik **BusyBox** ez badugu gure gailuan instalatuta. Instalatu egin nahi dugula hautatuko dugu.

5. Marketeko BusyBox-en orria agertuko zaigu, **Install** sakatu eta aplikazioak behar dituen baimenen berri emango zaigu, **Accept** sakatu.
6. Berriz ere aplikazioen menura joan eta droid **VNC server** ikonoan klik egin.
7. Arbeletik kopiatzeko ahalmenari buruz galdetuko digu, sakatu **Yes**.
8. droid VNC Server-en menua agertuko zaigu pantailan:
   - **Start Server** (Zerbitzaria martxan jartzeko)
   - **Stop Server** (Zerbitzaria geratzeko)
   - **Stopped** / **Started** (Bietako bat agertuko da, zerbitzariaren egoeraren berri ematen du)
9. Ezarpenak zehazteko **MENU** botoia sakatuko dugu eta bertan **Settings** klikatu.
10. Ezarpenak:
    - **VNC password**: Pasahitza zehazteko.
    - **Rotation**: Bezeroak ikusiko duen pantaila 0, 90, 180 edo 270 gradu bira dezakegu.
    - **Scale Screen**: Bezeroak ikusiko duen pantailaren tamaina zehatz daiteke, menuan aukera gehiago dauden arren %100 eta %50ek bakarrik funtzionatzen dute.
    - **VNC port**: Erabiliko den VNC ataka, lehenetsia: 5901.
    - **Start server on boot**: Zerbitzaria android gailua martxan jartzean abiaraziko den edo ez zehatz daiteke, balio lehenetsia ez da.
    - **Don't let screen turn off**: Pantaila itzal dadin galarazi, balio lehenetsia ez da.
    - **Notify when client connects**: Bezeroa konektatzen denean jakinerazpen bat agertuko da egoera barran, balio lehenetsia bai da.
    - **Disable Bottom Ad**: Pantailaren azpialdean agertzen den iragarkia kentzeko. Balio lehenetsia ez da.
11. VNC pasahitza idatzi ondoren atzera itzuli eta **Start Server** sakatuko dugu.

Arazoak izan ditut emulatzailean, ziur aski root baimenik ez dudalako:

- Ezin dut BusyBox instalatu, errorea ematen dit.
- Ezin dut droid VNC server abiarazi, errorea ematen dit. BusyBox falta delako, root baimenik ez dudalako?

Superuser aplikazioa ere instalatu dut baina ez dut lortu aurrerapenik:.

https://market.android.com/details?id=com.noshufou.android.su

5.10.7 android-vnc-server

Proba egin dut baina ez dut lortu martxan VNC zerbitzaria martxan jartzea, jarraibideak nahiko nahasiak baitira: http://code.google.com/p/android-vnc-server/wiki/README

Aplikazioa gailura kopiatu:

```
adb push /home/asier/Deskargak/androidvncserver /data
```

1526 KB/s (237144 bytes in 0.151s)

Baimenak egokitu:

```
adb shell chmod 755 /data/androidvncserver
```

Abiarazten saiatu:

```
adb shell /data/androidvncserver
```

Initializing framebuffer device /dev/graphics/fb0...

xres=240, yres=320, xresv=240, yresv=640, xoffs=0, yoffs=0, bpp=16

Initializing keyboard device /dev/input/event3 ...

cannot open kbd device /dev/input/event3

Ezin izan dut abiarazi, ondoren Android emulatzaileko Terminala ireki eta bertatik saiatu naiz aplikazioa abiarazten:

```
/data/androidvncserver &
```

Initializing framebuffer device /dev/graphics/fb0...

cannot open fb device /dev/graphics/fb0

5.10.8 Androidscreencast

1. Erabili aurretik beharrezkoa da android SDK, Java Runtime Environment 5 edo berriagoa instalatuta izatea.
2. Ondoren, konektatu android gailua ordenagailura edo emulatzailea martxan jarri eta detektatua izan dela egiaztatu “adb devices” erabiliz.
3. Fitxategi hau deskargatu:

   http://androidscreencast.googlecode.com/svn/trunk/AndroidScreencast/dist/androidscreencast.jnlp
4. `javaws <fitxategiaren_helbidea>` komandoa erabiliz abiarazten da.

   ```
   javaws /home/asier/Deskargak/androidscreencast.jnlp
   ```
5. Aplikazioaren sinadura digitala ezin izan dela egiaztatu dioen mezu bat agertuko zaigu, sakatu **Run**.
6. Leiho batean gure gailua edo emulatzailearen pantaila agertuko zaigu eta teklatua eta sagua erabiliz kontrolatzeko aukera izango dugu.
7. Menuak eskaintzen dituen aukerak:

   Record: MOV formatuan bideo bat graba dezakegu pantailaren edukiekin.

   Explore: Gure gailu edo emulatzailearen fitxategi-sisteman zehar nabigatzeko aukera ematen digu.

Open URL: URL bat idatzi eta OK sakatzean gailu/emulatzailearen nabigatzailean irekitzen du.

Sagua/teklatua ez badabiltza, ireki terminal bat eta egikaritu:

```
adb shell
su
chmod 777 /data/dalvik-cache
cd /data/dalvik-cache
chmod 777 ./
```

5.10.9 F-Droid

1. Lehenik Google Market-ekoak ez diren aplikazioak instalatzeko baimena behar dugu, horretarako:

   Aplikazioen menua > Settings > Applications > Unknown sources (Allow installation of non-Market applications) -> Onartu
2. http://f-droid.org/FDroid.apk deskargatu eta instalazioa abiarazi.
3. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, **Accept** sakatu.
4. Aplikazioak instalatzeko prozesua Android Market-ekoaren antzerakoa da.

5.10.10 Android Terminal Emulator

1. Google Market-ean Android Terminal Emulator aplikazioa aurkitu eta Install sakatu.
2. Aplikazioak behar dituen baimenen berri emango zaigu ondoren, Accept sakatu.
3. Android gailuan aplikazioen menuan Terminal Emulator-en ikonoan klik egingo dugu abiarazteko.

5.10.11 Bibliografia

ConnectBot

- http://tech.shantanugoel.com/2010/08/02/ssh-tunneling-android.html

SSH Tunnel

- http://lifehacker.com/5799888/ssh-tunnel-is-the-easiest-way-to-tunnel-on-your-android-device
- http://lifehacker.com/5803880/how-to-encrypt-all-internet-use-on-your-android-phone

Android-vnc-viewer

- http://code.google.com/p/android-vnc-viewer/wiki/Documentation

Droid-vnc-server

- http://opensourceexcedio.wordpress.com/2010/10/28/droid-vnc-server/

Android-vnc-server

- http://code.google.com/p/android-vnc-server/wiki/README

Root baimena lortzea

- http://stackoverflow.com/questions/5095234/how-to-root-getroot-access-on-android-emulator
- http://abd-tech.blogspot.com/2011/05/test-root-apps-on-android-emulator.html
- http://forum.xda-developers.com/showthread.php?t=682828

androidscreeencast

- http://code.google.com/p/androidscreencast/

6 Ondorioak: lortutakoa, baztertutakoa, epeak, etorkizuneko lanak

Orduen kontua:

Azalpena	Ordu kopurua
Port Knocking, Single Packet Authorization: Teoria aztertu eta knock/knockd eta fwknop-ekin probak egin	21
Wake on LAN: Teoria aztertu, makinak prestatu (Ubuntu USB batean instalatu, BIOSak egokitu, ...) eta probak egin.	16
DD-WRT: Teoria aztertu, bideratzailea flasheatu eta konfiguratu eta probak egin	13,5
SSH: Teoria aztertu, OpenSSH zerbitzaria instalatu eta konfiguratu eta probak egin	12
NX: Teoria aztertu, zerbitzariak (FreeNX eta neatx) eta bezeroak (QTNX eta OpenNX) instalatu eta probak egin	12
VNC: Teoria aztertu eta probak egin zerbitzari (Vino eta x11vnc) eta bezeroekin (Vinagre eta Firefox).	21,5
Android sistema eragilea aztertu, aplikazioen garapenerako tresnak instalatu (Android SDK, ADT plugina, ...), beharrezko gailu birtualak sortu emulatzailean erabiltzeko eta Market-a erabili ahal izateko egokitu, ...	23
Androiderako aplikazioen itzulpenak nola egin ikertu eta itzulpenak egin	17,25
Androiderako aplikazioak programatzearen oinarriak ikasten hasi	15
Androiderako aplikazioak aurkitu, aztertu eta probatu (bai emulatzailean bai benetako gailuan)	26
Komunikazioak: Tutorearekin (Roberto Cortiñas) eta lagundu didan lankide ohi batekin, e-posta eta telefonoa)	2
Memoria: Ikerketa lanean zehar harturiko oharrak txertatu, txukundu eta osatu. Sarrera, helburuak, diseinua eta ondorioak idatzi.	32,5
GUZTIRA	211,75

Beraz, guztira 211,75 ordu erabili ditut ikerketa lana osatzen. Hasiera batean zehaztutako 140/200 orduak gainditu ditut. Hasiberriaren akatsak egin ditut, planifikazio kaskarra egin nuen eta nahiko modu inprobisatuan garatu dut ikerketa. Helburuak zabalegiak eta ondo zehaztu gabeak ziren.

Egindako akatsetatik zerbait ikasi dut gutxienez proiektuen garapenean planifikazioak duen garrantziari buruz.

Hala ere, erabili nahi nituen teknika guztiak aztertu eta probatzeko modua izan dut. Port Knocking, Single Packet Authorization, urruneko administrazioa (SSH, SCP, VNC, NX), Androiderako aplikazioen itzulpengintza eta Android sistema eragilerako proiektuen garapenari buruz asko ikasi dut eta ikerketa lan hau, eta kurtsoa orokorrean, benetan esperientzia aberasgarria izan da niretzat.

Baztertutakoen artean:

- Ez dut Androiderako Fwknop erabiltzeko modurik izan Single Packet Authorization teknika lantzeko. Android 2.0 edo berriagoa behar du eta nik eskura nuen gailua 1.6 da.
- Androiderako VNC zerbitzaria ere ez dut probatzerik izan, root baimenak dituen gailu bat behar baita horretarako eta nik eskura nuenak ez zuen horrelakorik.
- SFTP probatzeko denborarik ez dit eman.
- Hasi aurretik ere zail ikusten nuen denbora ematea, baina helburuetako bat, aztertutako tekniketarako dauden Androiderako proiektu libreak batuko dituen aplikazio bat garatzea zen. Aurreikusi bezala denborarik ez dit eman eta Androiderako aplikazioen programazioa uste baino konplexuagoa iruditu zait. Etorkizuneko lanen atalean zehaztasun gehiago.

Ikerketa lana garatzen ari nintzela bururatutako ideiak, etorkizunean lantzeko interesgarri iruditutakoak:

- Zerbaiten garrantziaren denborarekiko aldaketaren estimazioa interneten gauzatzen diren bilaketen kopuruaren arabera. GPL lizentziadun Gootrude Graphs aplikazioa erabil daiteke horretarako(http://www.cipherdyne.org/gootrude).
- Port knocking-a malware eta botnet-etan erabiltzen al da?
- Suhesiei buruzko teoria eta inplementazioa ikertu.
- SPA paketeen detekzioa (Snort rules for SPA detection).
- Tor eta bere bateragarritasuna urruneko administrazioko teknikekin aztertu.
- VNC-k eskaintzen duen eguneraketen log-aren erabilpenak:
  - Erabileraren kontrola
  - Helburu forentseak
  - Erabilgarritasun azterketetan lagungarri
- VPN-ak (Virtual Private Network) ikertu
- SSH-k ordezkatu egin du Telnet. Baina FTP erabiltzen jarraitzen da SFTP edo SCP erabili ordez. Zergatik?
- Bi sistema eragileren artean zein abiarazi aukeratu ahal izatea WOL bidez ordenagailua piztean. Posible al da? Dirudienez sistema eragile lehenetsia bakarrik abiaraz daiteke.
- SSHFS (SSH File System) aztertu. http://en.wikipedia.org/wiki/SSHFS

- SFTP eta FTPS aztertu.
  - http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol
  - http://en.wikipedia.org/wiki/FTPS
- GNU/Linux edo Android erabiliz Windows eta MacOS makinak urrunetik administratzeko dauden aukera libreak aztertu. Eta alderantzizkoa ere bai, hau da, Windows edo MacOS makina bat erabiliz GNU/Linux administratzeko aukerak ere aztertu nahi nituzke.
- Android sistema eragilerako aplikazioen garatzaileentzat merkatu berriak irekitzen ari dira. Hasiera batean Android erabiltzen zuten gailu bakarrak mugikorrak baziren ere gero eta gailu mota desberdin gehiagok erabiltzen dute: tabletak, telebistak, urruneko agintegailuak, ... Etorkizunean merkatua oraindik gehiago zabalduko dela dirudi.
- Androiderako aplikazioen garapenean sakondu eta aztertutako Androiderako aplikazioak batetaratuko dituen aplikazio bat sortzea:
  - Makina piztu WOL bidez.
  - Ataka sekuentzia bidezko port knocking-a edo single packet authorization erabiliz beharrezko atakak ireki (adibidez, 22).
  - Ondoko teknikak erabiliz urruneko makinara konektatu eta beharrezko lanak burutu:
    - SSH
    - SCP
    - VNC konexio segurua SSH bidez birbidalita. (Beharrezko izanez gero VNC zerbitzaria martxan jarriz)
  - Makina itzali berriz ere WOL-erako prestatuta utzirik.

  Guzti hau ahalik eta modu errazenean egiteko aukera eskaini behar luke.
- Mugikor edo PDA bidez etxeko elektratresnak kontrolatzeko aukera.
- Android gailuetan root baimena nola lortu ikastea.
- Cyanogen-en mod-ak:

  http://www.cyanogenmod.com/

  http://en.wikipedia.org/wiki/CyanogenMod
- HierarchyViewer: Android aplikazioen garapenean lagungarri izan daiteke. Aplikazio honek emulatzaile/gailura konektatu eta uneko view-ari buruzko informazioa lortzea ahalbideratzen du: memorian dauden widget-ak, euren arteko erlazioak, propietateak, ... <android-sdk>/tools/hierarchyviewer
- Po fitxategiei buruz eta aplikazioen itzulpengintzari buruz ikertu. Eta bide batez zenbait aplikazio itzuli euskarara, adibidez, gWakeOnLan-ek testu gutxi dauka eta nahiko erraz itzultzeko modukoa dirudi: https://www.transifex.net/projects/p/gwakeonlan/
- xml2po tresna xml fitxategiak po formatura bihurtu eta errazago itzultzeko (gnome-doc-utils paketean dago).

- Androiderako aplikazio gehiago itzuli, adibidez:
  - Fwknop for android lokalizaziorako prestatu eta euskarara itzuli.
  - Android Terminal Emulator: https://github.com/jackpal/Android-Terminal-Emulator/wiki/Translating-to-Foreign-Languages
  - F-Droid bezeroa: http://f-droid.org/translate/projects/fdroidclient/
- Aisialdirako gailuen artean zabalduena da telebista, munduan zehar 4 mila milioi pertsonak ikusten dute. Duela gutxi kaleratutako ikerketa baten arabera, Euskadiko Autonomi Erkidegoan, pertsona bakoitzak bataz beste 229 minutu igarotzen ditu egunero telebista aurrean.

  Hala ere, konputagailuen eta mugikorren aisialdirako erabilera handituz doa, batez ere telebistak ez daukan ezaugarri bati esker, internet sarerako sarbidea. Interneti esker gustuko edukiak aurkitzea errazagoa bihurtu da.

  Bi esparru hauek, telebista eta internet, lotzeak ideia interesgarria dirudi beraz.
- Latex erabiltzen ikasi
- Bertsio kontrolerako teknikaren bat erabiltzen hasi (Git, Mercurial, ...)

Bukatzeko eskerrak eman nahi nizkieke ikerketa lan honetan nire tutore izan den Roberto Cortiñas-i eta bereziki nire lankide-ohi Alvaro Ricon-i eskainitako laguntzagatik eta bere ordenagailu, mugikor eta bideratzailearekin probak egiten uzteagatik.

Eskerrak baita graduondoko kurtsoan zehar irakasle eta ikaskide izan ditudan guztiei.

Asier Iturralde Sarasola

Usurbilen, 2011ko irailaren 12an

Eranskina: GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed,

as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the

Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and

publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has

been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and

will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or

of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009,

provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.3
> or any later version published by the Free Software Foundation;
> with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
> A copy of the license is included in the section entitled "GNU
> Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the
> Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.