VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Jussi Haapamäki

SULAUTETUN WEB-PALVELIMEN TIETOTURVA

Tekniikka ja liikenne
2012

VAASAN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

TIIVISTELMÄ

Tekijä	Jussi Haapamäki
Opinnäytetyön nimi	Sulautetun web-palvelimen tietoturva
Vuosi	2012
Kieli	suomi
Sivumäärä	48 + 1 liite
Ohjaaja	Jukka Matila

Opinnäytetyön tavoitteena oli tehdä tutkimustyö tietoturvallisesta sulautetusta web-palvelimesta. Tutkimustyö tehtiin Vaasan ammattikorkeakoululle, koska Vaasan ammattikorkeakoululla oli tarvetta tämän kaltaiselle tutkimukselle. Tutkimuksesta kehiteltiin myös tietoliikennetekniikan kursseilla käytettävä harjoitustyö. Sulautetun web-palvelimen kehitysalustana toimi Atmel Mature NGW100 Network Gateway Kit, jota käytetään sulautetulla Linux-käyttöjärjestelmällä.

Opinnäytetyö sisältää tutustumista NGW100-kehitysalustan tietoliikenteeseen, sulautetun Linux-käyttöjärjestelmän luomisen ja web-palvelimen tietoturvan rakentamisen. Tietoliikenne pitää sisällään reitityksen, DHCP-palvelimen ja siltauksen. Tietoturva pitää sisällään käyttäjäoikeudet, palomuurin sekä SSH- ja VPN-yhteydet.

Työn lopputuloksena sulautettu web-palvelin toimii kehitysalustalla, jolle saadaan luotua tietoturvallinen yhteys SSH-yhteydellä. Web-palvelimen ylläpitämälle html-sivustolle saadaan muodostettua yhteys VPN-yhteydellä. Web-palvelinta suojataan myös palomuurilla, joka estää ulkopuoliset vaarat.

Avainsanat	tietoturva, sulautettu web-palvelin, sulautettu Linux-käyttöjärjestelmä, VPN, SSH, palomuuri

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Tietotekniikan koulutusohjelma

ABSTRACT

Author	Jussi Haapamäki
Title	Embedded Web-Server Data Security
Year	2012
Language	Finnish
Pages	48 + 1 Appendices
Name of Supervisor	Jukka Matila

The purpose of the thesis was to research a data secure embedded web server. The research was made for Vaasa University of applied sciences, because Vaasa University of applied sciences needed this kind of research. The research also generated a practical work, which is used in telecommunication courses. The embedded web server's development board is Atmel Mature NGW100 Network Gateway Kit, which is used with an embedded Linux operating system.

The thesis gets acquainted with the telecommunication of the NGW100 development board, describes the making of an embedded Linux operating system, and the construction of a data secure web server. Telecommunication includes routing, DHCP server, and bridging. Data security includes user rights, firewall, and SSH and VPN connections.

The result of the thesis is an embedded web server that works on the development board, to which one can connect a data secure connection through SSH connection. A secure connection can be formed to the html site, maintained by the web server, through a VPN connection. The firewall also protects the web server from external threats.

Keywords data security, embedded web server, embedded Linux operating system, VPN, SSH, firewall

SISÄLLYS

LIITELUETTELO

1 JOHDANTO

Ennen oli käytäntönä se, että kun sulautettu alusta suoritti tehtäväänsä yrityksen sisällä, alustalle ei saatu muodostettua yhteyttä yrityksen yksityisen verkon ulkopuolelta. Tällöin tietoturvariskitkin olivat pienemmät. Tämä käytäntö on kuitenkin muuttunut laitteiden verkkomahdollisuuksien kehittyessä, minkä myötä myös etäyhteydet ovat lisääntyneet. Etäyhteydessä sulautettuun alustaan muodostetaan yhteys yksityisen verkon ulkopuolelta ja tällöin tietoturvan on oltava kunnossa. Tietoturvan avulla sulautettua alustaa suojataan ei-halutuilta osapuolilta.

Tämän opinnäytetyön tarkoituksena on tehdä tietotekniikan tutkimustyö Vaasan ammattikorkeakoululle, jossa yhdistyvät sulautetut järjestelmät ja tietoliikennetekniikka. Tutkimuksesta kehitellään myös tietoliikennetekniikan kursseilla käytettävä harjoitustyö. Työssä perehdytään sulautetun web-palvelimen tietoturvaan. Sulautetun web-palvelimen kehitysalustana toimii Atmel Mature NGW100 Network Gateway Kit. Kehitysalustalle luodaan sulautettu Linux-käyttöjärjestelmä, joka rakennetaan sellaisista Linux-järjestelmissä käytettävistä sovelluksista, joilla sulautetusta web-palvelimesta saadaan luotua tietoturvallinen.

Työssä sulautetun web-palvelimen ja asiakkaan välinen tietoturvallinen yhteys muodostetaan SSH- sekä VPN-yhteydellä. SSH ja VPN ovat ohjelmistoja, jotka salaavat kaiken kahden laitteen välillä tapahtuvan liikennöinnin. SSH:ssa yhteys suojataan julkisesta ja yksityisestä avaimesta muodostetulla avainparilla. Palvelinlaitteella hallussa oleva asiakkaan julkinen avain tulee sopia yhteen asiakaslaitteen hallussa olevan yksityisen avaimen kanssa. Lisäsuojauksena yksityinen avain suojataan vielä salasanalla, joten yksityistä avainta ei voida käyttää ilman salasanaa. VPN:ssä yhteys suojataan lähes samalla periaatteella, mutta kaksisuuntaisesti. Tämä tarkoittaa sitä, että molemmat osapuolet suorittavat todentamisen, näin ollen avaimiakin on enemmän.

2 TIETOTURVA

2.1 Tietoturva yleisesti

Aluksi on hyvä tietää mitä tietoturva on, miksi sitä käytetään ja minkälaisia tietoturvariskejä on olemassa. Tietoturva on yksi tärkeimmistä komponenteista verkkoyhteyksiä käyttäville laitteille, joita ovat mm. palvelimet ja työasemat. Laitteet on suojattava hyökkäyksiltä ja haittaohjelmilta, joilla laitteille pyritään luomaan harmia. Hyökkäyksen tai haittaohjelman kohteeksi joutuvan laitteen tärkeät tiedostot ovat erittäin suuressa vaarassa, jolloin pahimmassa tapauksessa koko laitteen toiminta on vaakalaudalla.

Esimerkiksi palvelinlaitteen tietoturvan tärkeys on helppo havainnollistaa verkkokaupassa asioidessa. Verkkokaupassa asioidessa, kaupan palvelimelle tallentuvat asiakkaan tiedot, kuten käyttäjätunnus, salasana, etu- ja sukunimi, osoite ja puhelinnumero. Tällöin jokainen asiakas haluaa olla varma siitä, että nämä tiedot eivät joudu vääriin käsiin. Näin ollen verkkokaupan palvelinlaitteen tietoturvan on oltava kunnossa.

2.1.1 Hyökkäykset

Verkkoyhteyksiä käyttäviä laitteita kohtaan on useita eri hyökkäysmahdollisuuksia. Näistä yleisimpiä ovat palvelunestohyökkäys ja mies välissä -hyökkäys. Palvelunestohyökkäys (DoS, Denial of Service) on palvelinlaitteelle kohdistuva hyökkäys, jolla ei haluta päästä sisään palvelimen järjestelmään, vaan sillä aiheutetaan harmia palvelimen toiminnalle. Palvelunestohyökkäyksessä palvelin ruuhkautetaan lähettämällä sille niin suuri määrä palvelupyyntöjä, että palvelimella ei ole enää resursseja palvella oikeita asiakkaitaan. /13/

Mies välissä -hyökkäyksellä (Man in the middle), hyökkääjä asettuu kahden osapuolen välille, esimerkiksi palvelimen ja työaseman välille. Tällöin hyökkääjä salakuuntelee osapuolien välistä liikennettä. Salakuuntelija, joka pääsee yhteyden kolmanneksi osapuoleksi, voi muuttaa osapuolien välistä liikennettä tai varastaa

tärkeää tietoa. Tämän ongelman ratkaisemiseen on kehitelty useita todentamista käyttäviä tietoturvasovelluksia.

2.1.2 Haittaohjelmat

Haittaohjelmat, eli virukset, ovat vaarallisia päästessään palvelimen tai työaseman sisälle. Haittaohjelman päästessä laitteen sisälle, se suorittaa järjestelmän toimintaa vaikeuttavia toimenpiteitä. Se voi esimerkiksi muokata tiedostoja tai poistaa niitä ja näin ollen haitata laitteen toimintaa. Pahimmassa tapauksessa koko laitteen toiminta voi olla vaakalaudalla haittaohjelman aiheuttamien tuhojen ansiosta. /14/

2.2 Linux ja tietoturva

Työssä laitteita käytetään eri Linux-järjestelmillä, joten Linux-järjestelmien pääasialliset tietoturvaominaisuudet on hyvä sisäistää. Työssä ovat vahvasti esillä pääkäyttäjä (root-käyttäjä) ja käyttäjäoikeudet. Pääkäyttäjä on Linux-järjestelmän ylläpitäjä, jolla on käyttäjäoikeudet koko järjestelmään. Tällöin pääkäyttäjää ei tule koskaan käyttää muissa kuin sellaisissa tilanteissa, joissa sitä oikeasti tarvitaan, esim. ohjelmistojen asentamisessa ja oikeuksien jakamisessa. Järjestelmään tulee luoda ns. normaaleja käyttäjiä, joilla järjestelmää käytetään arkipäiväisissä tehtävissä. Normaaleilla käyttäjillä on käyttäjäoikeudet vain tiettyihin hakemistoihin ja tiedostoihin. Näitä oikeuksia pääse muokkaamaan vain pääkäyttäjä. Pelkästään jo tällä ominaisuudella järjestelmän tietoturvatasoa saadaan nostettua.

Esimerkiksi, jos normaali käyttäjä Liisa saa epäonnekseen hankittua haittaohjelman omiin tiedostoihinsa, niin haittaohjelma aiheuttaa harmia Liisan tiedostoissa ja järjestelmä on vielä pelastettavissa. Tapauksessa, jossa haittaohjelman hankintavaiheessa oltaisiin käytetty pääkäyttäjää, niin harmia aiheutuisi koko järjestelmälle ja näin ollen järjestelmän toiminta olisi vaarassa. /10/

2.2.1 SELinux ja GRSecurity

Tiedostojen ja sovellusten tarkempien käyttäjäoikeuksien valvontaa varten on kehitelty työkalut SELinux ja GRSecurity. SELinux on NSA:n (National Security Agency) kehittelemä tietoturvaa lisäävä työkalu, jolla tiedostojen ja sovelluksien

käyttäjäoikeuksia saadaan määriteltyä tarkemmin kuin normaalisti käytettävillä Linux-ominaisuuksilla. GRSecurity on SELinuxin kilpailija. GRSecurityn kehittäjän mukaan, se mahdollistaa vielä tarkempien käyttäjäoikeuksien hallitsemisen kuin SELinux. /10/

2.2.2 Palomuuri ja TCP-wrapper

Palomuuria käytetään suodattamaan tietoliikennettä kahden eri verkon välillä. Palomuuri voidaan määritellä joko palvelinlaitteelle tai työasemalle. Palvelimen palvellessa esimerkiksi suurta aliverkkoa, siinä on syytä olla käytössä jonkin tason palomuuri. Pääasiassa palomuurilla estetään ulkoverkosta tulevat yhteydenotot ja aliverkosta suoritettavat yhteydenotot sallitaan vain tiettyjen palvelimen porttien kautta. Linuxissa yksinkertainen palomuuri kantaa nimeä Netfilter, jonka asetuksia määritellään iptables-työkalulla. /9/

TCP-wrapper on ikään kuin palomuuri, mutta sillä saadaan asetettua tarkemmin verkkoyhteyksissä käytettävät suodatukset. TCP-wrapper -työkalulla voidaan määritellä mm. käyttäjä, joka voi ottaa palvelimen kautta verkkoyhteyden ulkoiseen verkkoon. /10/

2.3 Työssä käytetty tietoturva

Työssä tietoturvallinen yhteys muodostetaan sulautetulle web-palvelimelle. Palvelin toimii sulautetulla alustalla, joka oikeassa tilanteessa voisi suorittaa tärkeitä tehtäviä, kuten valvoa hälytyksiä, suorittaa lämpötilamittauksia tai ohjata teollisuuslaitteita. Tällöin kun palvelimelle ollaan yhteydessä, on varmistuttava siitä, että yhteys on tietoturvallinen. Tietoturvallisien yhteyksien luomiseen työssä käytetään SSH- sekä VPN-yhteyksiä. Näissä yhteyksissä käytetään todentamista julkisesta ja yksityisestä avaimesta muodostetulla avainparilla. Todentamisella saadaan torjuttua mm. aiemmin mainittuja mies välissä -hyökkäyksiä. Tietoturvallisen yhteyden muodostamisen lisäksi palvelimelle pystytetään palomuuri, jolla estetään kaikki muut palvelimelle tulevat palvelupyynnöt, paitsi SSH- ja VPN-pyynnöt.

2.3.1 SSH

SSH-yhteydellä (Secure Shell) mahdollistetaan se, että asiakas pääse etäyhteydellä kiinni palvelimen järjestelmään. SSH-yhteyttä muodostettaessa tietoturvaominaisuuksista käytetään todentamista ja käyttäjäoikeuksia. Yhteydessä olevien osapuolien todentaminen toteutetaan julkisesta ja yksityisestä avaimesta muodostuvalla avainparilla. Todentamisella tarkistetaan se, että yhteyttä ottava osapuoli on luotettava osapuoli. SSH-yhteyttä muodostettaessa otetaan huomioon myös se, että yhteyttä ei muodosteta palvelimen pääkäyttäjään. Palvelimella on oltava luotuna ns. normaali käyttäjä. Yhteyttä ei muodosteta pääkäyttäjään, koska pääkäyttäjällä on täydet oikeudet koko järjestelmään. Pääkäyttäjällä voidaan aiheuttaa paljon enemmän harmia järjestelmässä kuin normaalilla käyttäjällä.

2.3.2 VPN

VPN-yhteydellä (Virtual Private Network) mahdollistetaan tietoturvallinen tietoliikenneyhteys asiakkaan ja palvelimen välille. VPN-yhteydellä ei ole SSH:n tavoin tarkoitus ottaa yhteyttä palvelimen järjestelmään, vaan sillä luodaan tietoturvallinen keskusteluyhteys asiakkaan ja palvelimen välille. VPN luo yhteydessä oleville osapuolille virtuaaliset verkkokortit, joiden kautta osapuolet keskustelevat keskenään. Osapuolien todentaminen suoritetaan lähes samalla periaatteelle kuin SSH:ssa, mutta kaksisuuntaisesti. Tällä tarkoitetaan sitä, että molemmat osapuolet varmistavat vastapuolen luotettavuuden, näin ollen avaimiakin on enemmän.

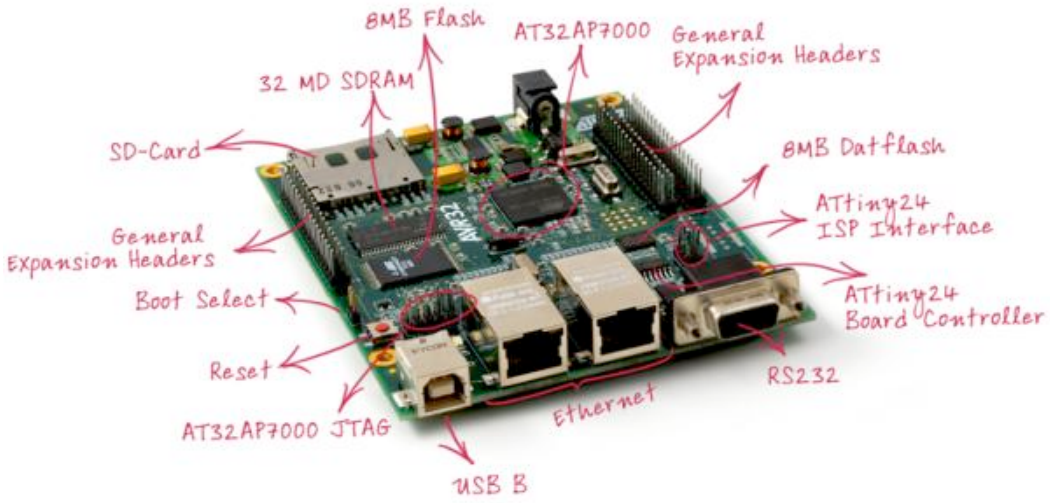
2.3.3 Palomuuri

SSH- ja VPN-etäyhteyksiä käytettäessä, sulautetulla alustalla oleva web-palvelin on yhteydessä julkiseen verkkoon. Laitteella, joka on suoraan yhteydessä julkiseen verkkoon, on oltava palomuuri. Palomuurilla estetään julkisesta verkosta tulevat hyökkäykset. Palomuuri pystytetään sulautetun alustan julkisen verkon puoleiselle portille, joka määritellään estämään kaikki muut yhteydenottopyynnöt, paitsi SSH- ja VPN-pyynnöt. Palomuurilla sallitaan myös kaikki VPN-yhteydessä pystytetyn virtuaalisen verkkokortin kautta tulevat pyynnöt. Näillä estoilla sulautettuun web-palvelimeen ei saada luotua muita kuin tietoturvallisia yhteyksiä.

3 KÄYTETTY KEHITYSALUSTA

3.1 Atmel Mature NGW100 Network Gateway Kit

NGW100 Network Gateway Kit on Atmelin luoma kehitysalusta, joka on erityisesti luoto Linux-pohjaisien verkkosovellusten kehitys- ja tutkimisympäristöksi. Kehitysalusta on varustettu suurella määrällä järjestelmämuistia ja siltä löytyy laajat liitännät, kuten kaksi Ethernet-porttia (WAN ja LAN), RS232- ja USB 2.0-liitännät ja MMC/SD-korttipaikka. Kehitysalusta ja sen liitännät on esiteltynä kuvassa yksi. Laajojen liitäntöjen ansiosta kehitysalusta on myös ihannekehitysympäristö AT32AP7000-prosessorille. Kehitysalustalta löytyy myös suuri määrä I/O-liitäntöjä, joita voidaan hyödyntää omien sovelluksien kehittämisessä. /1/



Kuva 1. Atmel NGW100 -kehitysalusta ja sen liitännät /5/

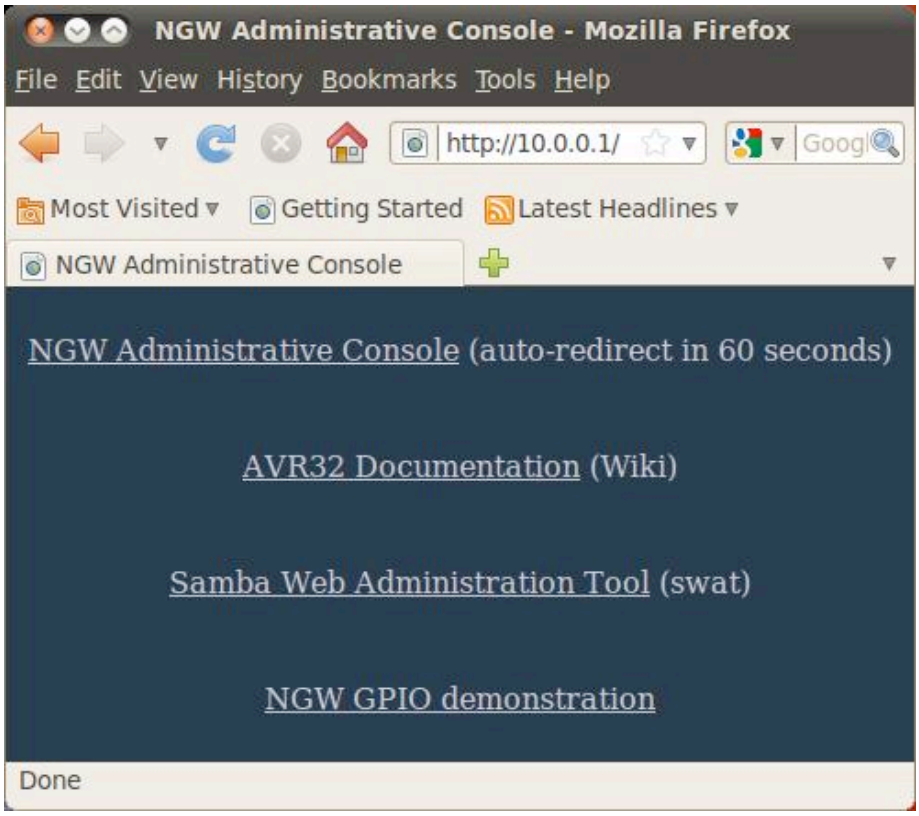
3.2 Kehitysalustaan tutustuminen

Työssä lähdettiin tutustumaan NGW100-kehitysalustan käyttöön ja ominaisuuksiin ensikertalaisen näkökulmasta. Tutustuminen tapahtui AVR Freaks Wikin avulla, joka käytännössä toimii myös kehitysalustan käyttöohjeena. Wikistä löytyvät hyvät ohjeet kehitysalustan parissa työskentelevälle ensikertalaiselle kuin myös kokeneemmallekin käyttäjälle. AVR Freaks Wiki löytyy osoitteesta http://www.avrfreaks.net/wiki/index.php/Documentation:NGW.

Työ aloitetaan tutustumalla kehitysalustan Ethernet-liitäntöihin, MMC/SD-korttipaikkaan ja RS232-sarjaporttiin. Näiden liitäntöjen hallitseminen on suuri osa työn suorittamista, siksi niihin on syytä perehtyä ensimmäisenä.

3.2.1 Kehitysalustan käyttäminen reitittimenä

NGW100-kehitysalustalla on kaksi Ethernet-liitäntää, WAN (eth0) ja LAN (eth1). Reititin kytkennässä WAN-liitäntä on kytketty ulkoiseen verkkoon ja LAN-liitäntä työasemalle. Tällä kytkennällä kehitysalusta toimii reitittimenä oletusasetuksilla, jolloin WAN-portti saa verkkoasetuksensa ulkoverkosta ja LAN-portin verkkoasetukset ovat kiinteät. Kiinteissä verkkoasetuksissa LAN-portin IP-osoiteeksi on määritelty 10.0.0.1. Kehitysalustan oletusreititin sisältää DHCP-palvelimen. DHCP:n IP-osoitealue on oletuksena 10.0.0.20-254, joten työaseman IP-osoite asettuu tälle välille, kunhan työasema on asetettu hakemaan verkkoasetukset automaattisesti. Reitittimen oletusasetuksia päästään halutessa muokkaamaan sivustolta http://10.0.0.1, joka näkyy kuvassa kaksi. /6/

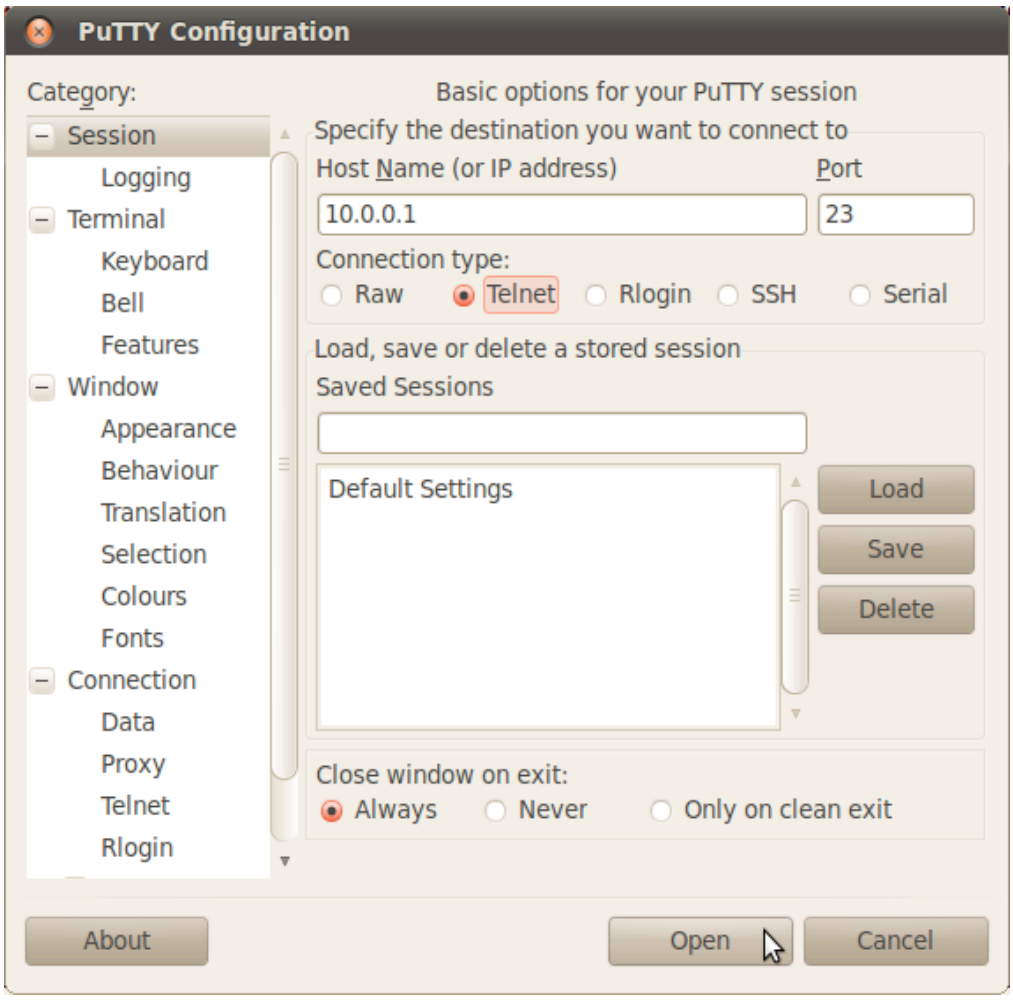


Kuva 2. NGW100-kehitysalustan asetusten hallinta

3.2.2 SD-kortin käyttäminen ulkoisena tallennusvälineenä

NGW100-kehitysalustalla voidaan käyttää USB-massamuistilaitetta ulkoisena tallennusvälineenä. Tässä työssä USB-massamuistilaitteena käytetään SD-korttia. Kehitysalusta pitää olla kytkettynä USB-kaapelilla työasemaan ja SD-kortti liitettynä alustalla olevaan MMC/SD-korttipaikkaan. Jotta SD-kortti saadaan näkyviin työasemalle, pitää se aluksi asettaa päälle. Tämä operaatio suoritetaan RS232-sarjaportin kautta telnet-yhteydellä. /4/

Telnet-yhteyden muodostamiseen käytetään Putty-terminaaliohjelmaa. Puttyn asetuksista rastitetaan kohta telnet ja IP-osoitteeksi asetettaan 10.0.0.1, eli kehitysalustan LAN-portti. Puttyn asetukset näkyvät kuvassa kolme.



Kuva 3. Putty-terminaaliohjelman asetukset

Kehitysalustalla oltaessa kiinni telnet-yhteydellä, SD-kortti saadaan näkyviin työasemalle Puttyn komentoriviltä komennolla: /4/

```
usb-mass-storage on
```

Komennon jälkeen SD-kortti ilmestyy näkyville työasemalle ja Putty ilmoittaa SD-kortin olevan käytettävissä. Puttyn ilmoitus näkyy kuvassa neljä.

```
BusyBox v.1.4.2 (2007-04-17 15:34:55 CEST) Built-in shell (ash)
Enter ‘help’ for a list of built-in commands.

~ # usb-mass-storage on
USB: on
```

Kuva 4. Puttyn ilmoitus, SD-kortti on käytettävissä

4 SULAUTETTU LINUX-KÄYTTÖJÄRJESTELMÄ

4.1 Buildroot

Työssä luodaan sulautettu Linux-käyttöjärjestelmä, jolla NGW100-kehitysalustaa käytetään. Sulautettuun Linux-käyttöjärjestelmään tarvitaan kaksi komponenttia, käyttöjärjestelmän ydin (kernel) ja juuritiedostojärjestelmä (root filesystem). Näiden komponenttien luomiseen erityisen hyvä työkalu on Buildroot. /7/

Buildroot on työkalu, joka koostuu joukosta makefile- ja patch-tiedostoja. Näiden tiedostojen avulla käyttöjärjestelmän komponentit haetaan, rakennetaan ja käännetään. Näiden vaiheiden lopputuloksena, Buildroot luo binääritiedoston käyttöjärjestelmän ytimestä ja pakkaa juuritiedostojärjestelmän tar-tiedostoksi. Buildrootista löytyy graafisia valikoita, joiden avulla käyttöjärjestelmään voidaan tehdä erilaisia valintoja. Valikoiden avulla käyttöjärjestelmään saadaan valittua käytettävä ydin, ytimelle tarvittavat ajurit ja erilaisia Linux-käyttöjärjestelmissä käytettäviä sovelluksia. Monenlaisten ohjelmistojen avulla NGW100-kehitysalustalle saadaan rakennettua halutunlainen käyttöjärjestelmä. /7/

4.2 Sulautetun Linux-käyttöjärjestelmän luonti Buildrootilla

Työssä käytettiin Buildrootin versiota 2011.05. Kyseinen versio on saatavilla sivustolta http://buildroot.uclibc.org/downloads/. Sivustolta ladattava tar-tiedosto puretaan haluttuun paikkaan työasemalle. Komentorivillä siirrytään purettuun hakemistoon, eli buildroot-hakemiston juureen ja syötetään komento:

```
make atngw100_defconfig
```

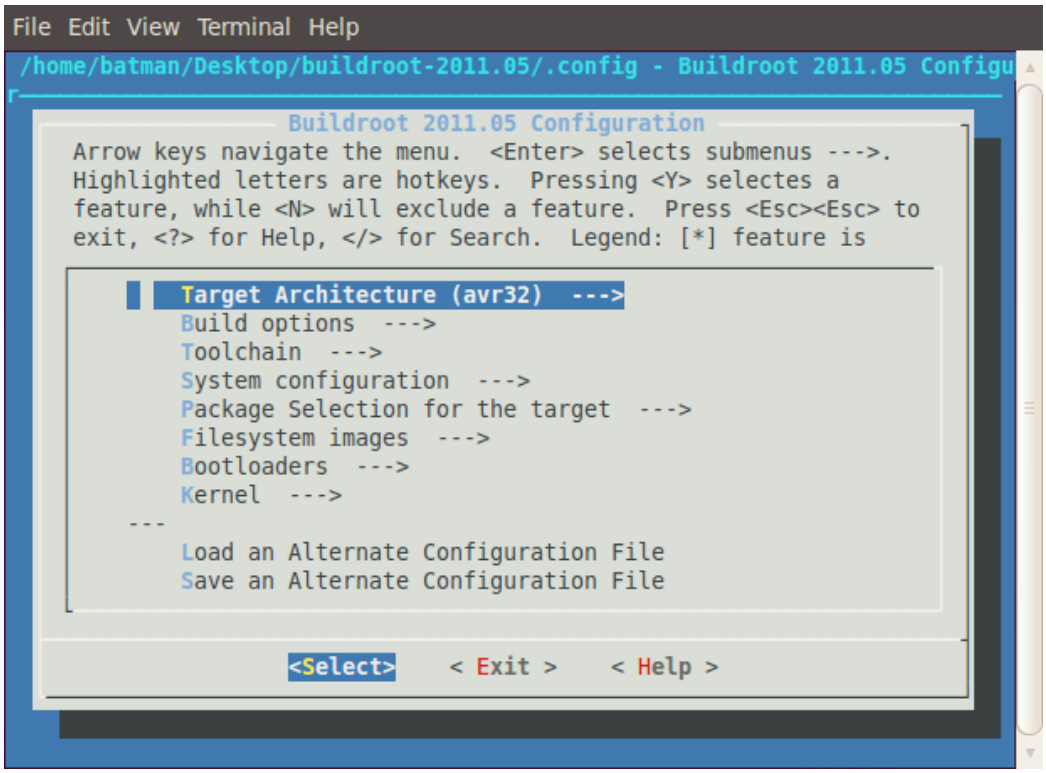
Tällä komennolla Buildroot lataa NGW100-kehitysalustan oletusasetukset ja tallentaa ne .config-tiedostoon. Komennon mentyä onnistuneesti läpi, Buildroot ilmoittaa kirjoittaneensa asetukset .config-tiedostoon. /7/

4.2.1 Asennettavien sovelluksien valinta

Luotavaan Linux-käyttöjärjestelmään pystytään valitsemaan asennettavaksi erilaisia ohjelmistoja ja sovelluksia graafisen menuconfig-valikon avulla. Menuconfig-

valikko on esiteltynä kuvassa viisi. Valikon jokaista valittavissa olevaa ohjelmistoa vastaa yksi makefile ja yksi patch-tiedosto. Patch-tiedosto kertoo polun sivustolle, josta ohjelmisto haetaan ja makefile-tiedostolla ohjelmisto rakennetaan ja käännetään. Menuconfig-valikkoon siirryttäessä, on komentorivillä siirryttävä buildroot-hakemiston juureen ja hakemiston juuressa syöttetään komento: /3/

```
make menuconfig
```



Kuva 5. Buildroot menuconfig-valikko

Menuconfig-valikosta löytyy laaja valikoima erilaisia Linux-järjestelmissä käytettäviä ohjelmistoja. Työtä varten menuconfig-valikossa on tehtävä kaksi muutosta asennettavien ohjelmistojen osalta. Ensimmäisenä on vaihdettava järjestelmässä käytettävä ydin. Buildrootin versiossa 2011.05 on oletuksena valittuna ytimen vanha versio. Vanhan version polku ei ole enää voimassa, joten asennusvaiheessa Buildroot ei saa haettua ydintä verkosta. Näin ollen järjestelmään on valittava asennettavaksi ytimestä uudempi versio, jonka polku on voimassa. Työssä käytettiin versiota 2.6.38.7 (Kernel → Kernel version → 2.6.38.7). Toisena muutoksena

Linux-järjestelmän on tuettava ext2-tiedostojärjestelmää (Filesystem images → ext2 root filesystem). Tämä sen takia, että järjestelmä siirretään SD-kortille ja SD-kortti alustetaan käyttämään ext2-tiedostojärjestelmää. Tämä on SD-korteissa yleisesti käytetty tiedostojärjestelmä. Muiden tiedostojärjestelmien tukeminen voidaan jättää pois, koska niitä ei tarvita.

Järjestelmään lisätään myös muutama ohjelmisto, joiden avulla tutkitaan kehitysalustan tietoliikennettä ja tietoturvaa. Työssä suoritettavia yhteyksiä varten, järjestelmään valittiin asennettavaksi seuraavat ohjelmistot:

- nano

  Nano on yksinkertainen Linux-järjestelmissä käytettävä tekstieditori, jolla on jatkossa helppo tehdä tarvittavat muutokset tekstitiedostoihin. (Polku: Package Selection for the target → Text editors and viewers → nano)
- dnsmasq

  Dnsmasq on kevyt ja helposti määriteltävissä oleva DHCP-palvelinohjelmisto. (Polku: Package Selection for the target → Networking applications → dnsmasq)
- iptables

  Iptables on ohjelmisto, jolla määritellään kehitysalustan porttien verkkoliikenne, eli reititys ja palomuuri. Iptablesilla määritellään siis se, mikä verkkoliikenne on sallittua ja mikä ei. (Polku: Package Selection for the target → Networking applications → iptables)
- Enable large file (files > 2 GB) support

  Enable large file (files > 2 GB) support on iptablesin vaatima lisäohjelmisto. (Polku: Toolchain → Enable largefile (files > 2GB) support).
- bridge-utils

  Bridge-utils on ohjelmisto, jolla järjestelmään luodaan siltaava yhteys. (Polku: Package Selection for the target → Networking applications → bridge utils)

- lighttpd

  Lighttpd on ohjelmisto, jolla saadaan luotua kevyt, turvallinen ja joustava web-palvelin. (Polku: Package Selection for the target → Networking applications → lighttpd)
- openssh

  Openssh on ohjelmisto, jolla luodaan tietoturvallinen SSH-yhteys web-palvelimen ja asiakkaan välille. (Polku: Package Selection for the target → Networking applications → openssh)
- openvpn

  Openvpn on ohjelmisto, jolla luodaan tietoturvallinen VPN-yhteys web-palvelimen ja asiakkaan välille. (Polku: Package Selection for the target → Networking applications → openvpn)
- Enable RPC support

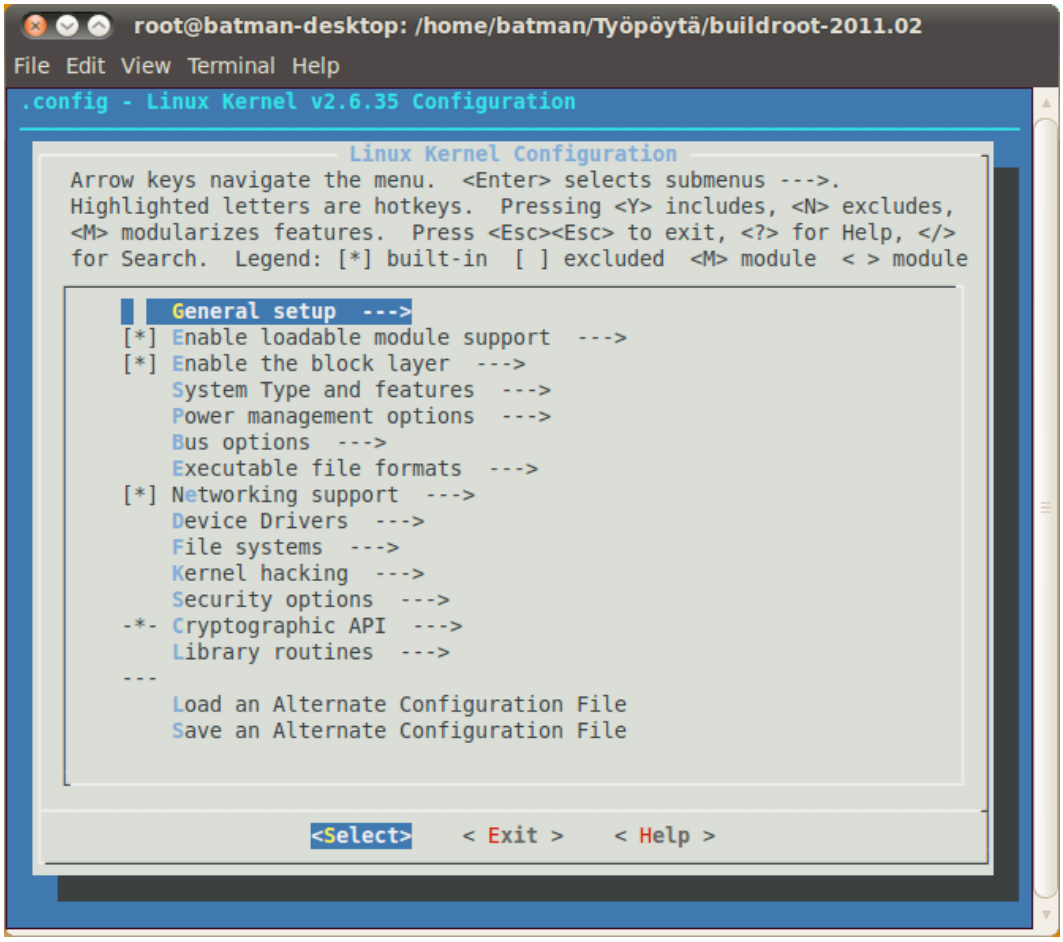
  Enable RPC support on lisäohjelmisto, jonka openvpn vaatii. (Polku: Toolchain → Enable RPC support)

Menuconfig-valikossa tehdyt muutokset ja lisäykset tallennetaan poistuttaessa menuconfig-valikosta. Tallennuksen jälkeen Buildroot kirjoittaa valinnat .config-tiedostoon. Tästä tiedostosta Buildroot lukee, mitkä sovellukset sen tulee hakea ja asentaa käännösvaiheessa.

4.2.2 Ajureiden valinta

Buildrootin toinen tärkeä valikko on linux-menuconfig, jossa pystytään muokkaamaan ytimen ominaisuuksia. Linux-menuconfig -valikko on esiteltynä kuvassa kuusi. Tässä valikosta määritellään järjestelmän ytimen ajurit, eli laiteajurit. Valikkoa käytetään mm. sellaisissa tilanteissa, joissa kehitysalustaan liitetään lisälaitteita. Valikkoon päästään syöttämällä buildroot-hakemiston juuressa komento:

```
make linux-menuconfig
```



Kuva 6. Buildroot linux-menuconfig -valikko

Valikosta on oletuksena valittuna työssä tarvittavat ajurit, mutta yksi ajuri on kuitenkin vielä lisättävä. Tämä ajuri on Virtual terminal (Polku: Device Drivers → Character devices → Virtual terminal). Ajurilla saadaan poistettua virheilmoitus:

can't open /dev/tty1: No such device or directory.
can't open /dev/tty2: No such device or directory.
can't open /dev/tty3: No such device or directory.
…

Ilman Virtual terminal -ajuria käyttöjärjestelmä käynnistyy SD-kortilta muuten normaalisti, mutta järjestelmä tulostaa komentoriville jatkuvasti ylläpuolella näkyvää virheilmoitusta.

4.2.3 Käyttöjärjestelmän kääntäminen

Sulautettu Linux-käyttöjärjestelmä on sovelluksien ja ajureiden valinnan jälkeen valmis käännettäväksi. Ennen kääntämistä on kuitenkin varmistettava, että kehityskoneella on asennettuna seuraavat sovellukset: /3/

- C compiler (GCC)
- C++ compiler for Qtopia (G++)
- GNU make
- subversion
- sed
- flex
- bison
- patch
- gettext
- libtool
- texinfo
- autoconf
- automake
- ncurses library (development install)
- zlib library (development install)
- libacl library (development install)
- lzo2 library (development install)

Kun kehityskoneelta löytyy vaadittavat sovellukset, järjestelmä käännetään Buildroot hakemiston juuresta komennolla: /3/

```
make
```

Buildroot asettaa käännöksen lopputuloksen output-hakemistoon, joka ilmestyy onnistuneen käännöksen jälkeen buildroot-hakemiston juureen. Output-hakemistosta löytyy images-hakemisto, jonne Buildroot on luonut järjestelmän ytimen ja juuritiedostojärjestelmän. Ytimen binääritiedoston nimeksi on asettunut

uImage ja juuritiedostojärjestelmä on pakattu rootfs.tar -tiedostoksi. Käyttöjärjestelmä on nyt valmis siirrettäväksi SD-kortille.

4.3 SD-kortin alustus ja ext2-tiedostojärjestelmän luonti

Ennen järjestelmän siirtämistä SD-kortille, SD-kortti alustetaan ja sille luodaan SD-korteissa yleisesti käytetty etx2-tiedostojärjestelmä. Kortti alustetaan fdisk-työkalulla, joka sisältää itsessään ohjeet, joiden avulla alustaminen onnistuu. Fdisk-työkalu käynnistetään komennolla *fdisk /dev/sdd*. Kortin osiot saadaan näkyville komennolla *p* (print the partition table). Kortilta poistetaan kaikki osiot komennolla *d* (delete a partition), jolloin fdisk valitsee automaattisesti poistettavaksi osion yksi, jos kortilla ei ole muita osioita. Uusi osio luodaan komennolla *n* (add a new partition) ja osioksi valitaan primary partition (ensiöosio) komennolla *p*. Työssä korttia ei tarvitse jakaa useaan osioon, joten valitaan osioksi numero yksi ja sille käyttöön kortin koko kapasiteetti. Tehdyt asetukset kirjoitetaan kortille komennolla *w* (write table to disk and exit). Nämä vaiheet on suoritettu kuvassa seitsemän. /2/

```
user@user-desktop:~$ sudo su
root@user-desktop:~# umount /dev/sdd1
root@user-desktop:~# fdisk /dev/sdd

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): p

Disk /dev/sdd: 253 MB, 253231104 bytes
16 heads, 32 sectors/track, 966 cylinders
Units = cylinders of 512 * 512 = 262144 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Device Boot      Start         End      Blocks   Id  System
/dev/sdd1               1         966      247280   83  Linux

Command (m for help): d
Selected partition 1

Command (m for help): n
Command action
e   extended
p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-966, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-966, default 966):
Using default value 966

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

root@user-desktop:~#
```

Kuva 7. SD-kortin alustus ja uudelleenosiointi

Ext2-tiedostojärjestelmän luonti suoritetaan komennolla *mke2fs /dev/sdd1*. Tämä vaihe näkyy kuvassa kahdeksan. /2/

```
root@user-desktop:~# mke2fs /dev/sdd1
mke2fs 1.41.11 (14-Mar-2010)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
62000 inodes, 247280 blocks
12364 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
31 block groups
8192 blocks per group, 8192 fragments per group
2000 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729, 204801, 221185

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
root@user-desktop:~#
```

Kuva 8. Ext2-tiedostojärjestelmän luonti

4.4 Käyttöjärjestelmän siirto SD-kortille

Käyttöjärjestelmää varten luodaan uusi hakemisto samaan paikkaan, jossa buildroot-hakemisto sijaitsee. Hakemiston luodaan samaan paikkaan buildroot-hakemiston kanssa selkeyden vuoksi. Hakemiston nimeksi annetaan sd-kortti, joka kuvaa hakemiston käyttötarkoitusta. SD-kortti ja sd-kortti -hakemisto liitetään toisiinsa, jolloin kaikki tiedostot, jotka siirretään sd-kortti -hakemistoon, siirtyvät myös SD-kortille. Liitäntä suoritetaan komennolla: /2/

```
mount /dev/sdd1 …/sd-kortti
```

Järjestelmän juuritiedostojärjestelmä (rootfs.tar) puretaan sd-kortti -hakemistoon. Komentorivillä on oltava sd-kortti -hakemistossa ja purkaminen suoritetaan komennolla: /2/

```
tar -xf …/buildroot-2011.05/output/images/rootfs.tar
```

Komentorivillä pysytään edelleen sd-kortti -hakemistossa ja hakemistoon luodaan boot-alihakemisto. Alihakemistoon kopioidaan järjestelmän ytimen binääritiedosto (uImage) komennolla:

```
cp …/buildroot-2011.05/output/images/uImage boot
```

Ytimen binääritiedosto on sijaittava boot-hakemistossa, koska käyttöjärjestelmän käynnistysvaiheessa kehitysalusta ohjataan käynnistämään ydin boot-hakemistosta.

4.5 Käyttöjärjestelmän käynnistys SD-kortilta

Käyttöjärjestelmän käynnistys suoritetaan telnet-yhteydellä. Järjestelmän käynnistys SD-kortilta vaatii sen, että kehityskoneella on telnet-yhteyden muodostamista varten asennettuna jokin terminaaliohjelma. Linuxissa hyviä terminaaliohjelmia ovat esim. GtkTerm ja Minicom. Terminaaliohjelmassa pitää olla valittuna seuraavat asetukset:

Port:	/dev/ttyS0
Speed:	115200
Parity:	none
Bits:	8
Stopbits:	1
Flowcontrol:	none

Terminaaliohjelman on hyvä olla käynnissä ennen SD-kortin liittämistä kehitysalustalle. SD-kortti liitetään kehitysalustalle ja alustalle kytketään virta päälle. Kehitysalusta käynnistyy automaattisesti sen omalla muistilla olevalla oletusjärjestelmällä ja käynnistys saadaan pysäytettyä painamalla välilyöntiä kohdassa:

```
Press SPACE to abort autoboot in 1 seconds
```

Välilyöntiä painamalla päästään määrittelemään kehitysalustan U-bootin asetukset oikeiksi. U-boot on kehitysalustan prosessorilta löytyvä ohjelma, jolla saadaan määriteltyä miltä laitteelta kehitysalusta käynnistyy. U-boot määritellään käynnis-

tämään käyttöjärjestelmän ydin SD-kortilta. SD-korttia kuvaa laitetunnus mmcblk0p1. Tässä vaiheessa on myös kerrottava mitä tiedostojärjestelmää SD-kortti käyttää. U-bootin määritykset näkyvät kuvassa yhdeksän.

```
Uboot> askenv boocmd
Please enter ‘bootcmd’:mmcinit; ext2load mmc 0:1 0x10400000 /boot/uImage; bootm
Uboot> set bootargs ‘console=ttyS0 root=/dev/mmcblk0p1 rootwait’
Uboot> saveenv
Uboot> boot
```

Kuva 9. U-boot märitykset, kehitysalustan käynnistäminen SD-kortilta

Kehitysalusta on nyt käytettävissä luodulla käyttöjärjestelmällä ja järjestelmän hakemistorakenteessa pystytään liikkumaan normaaleilla Linux-komennoilla.

Tilanteissa, joissa kehitysalustaa ei ole tarvetta enää käyttää SD-kortilta, vaan kehitysalustaa halutaan käyttää oletusjärjestelmällä, alusta käynnistetään U-bootista kuvassa kymmenen näkyvillä määrityksillä. Kehitysalustan oma muisti on Flash-muistia ja käyttää jffs2-tiedostojärjestelmää, joten kehitysalusta on määriteltävä käynnistymään Flash-muistilta ja U-bootissa on myös kerrottava, että Flash-muisti käyttää jffs2-tiedostojärjestelmää. Flash-muistia kuvaa laitetunnus mtdblock1.

```
Uboot> askenv bootcmd
Please enter 'bootcmd':fsload; bootm
Uboot> set bootargs ‘console=ttyS0 root=/dev/mtdblock1 rootfstype=jffs2’
Uboot> saveenv
Uboot> boot
```

Kuva 10. U-boot määritykset, kehitysalusta käynnistäminen sen omalta muistilta

5 TIETOLIIKENNE

5.1 DHCP-palvelin ja reititys

NGW100-kehitysalustan tietoturvaa tutkittaessa, on aluksi hyvä tietää kehitysalustan perustietoliikennemahdollisuuksista. Alustan tietoliikenteeseen tutustuminen suoritetaan määrittelemällä alusta reitittäväksi DHCP-palvelimeksi (Dynamic Host Configuration Protocol). Tämä suoritetaan asennettujen dnsmasq- ja iptables-ohjelmistojen avulla.

Kehitysalustan porteille määritellään verkkoasetukset. Verkkoasetukset määritellään interfaces-tiedostossa, joka sijaitsee hakemistossa /etc/network. Ulkoverkon puoleinen portti (eth0) asetetaan hakemaan verkkoasetukset automaattisesti ja sisäverkon puoleisen portin (eth1) verkkoasetukset asetetaan kiinteiksi. Työssä käytettiin kuvassa 11 näkyviä verkkoasetuksia.

```
# Configure Loopback
auto lo
iface lo inet loopback
# Eth0-portin määritykset. Portti asetetaan automaattisesti päälle.
auto eth0
# Määritellään automaattiset verkkoasetukset.
iface eth0 inet dhcp
# Eth1-portin määritykset. Portti asetetaan automaattisesti päälle.
auto eth1
# Määritellään kiinteät verkkoasetukset.
iface eth1 inet static
        address 10.0.0.1
        network 10.0.0.0
        netmask 255.255.255.0
        broadcast 10.0.0.255
```

Kuva 11. Kehitysalustan verkkoasetukset, interfaces-tiedosto

Kehitysalustan verkkoasetuksien ollessa kunnossa, määritellään DHCP-palvelimen asetukset. Työssä DHCP-palvelin määritellään jakamaan IP-osoitteita väliltä 10.0.0.2-254. DHCP:n asetukset määritellään dnsmasq.conf -tiedossa, joka tulee sijaita hakemistossa /etc. Kuvassa 12 näkyvät työssä käytetyt DHCP-palvelimen asetukset.

```
# DHCP-palvelimena toimii eth1-portti.
interface=eth1
# DHCP-palvelimen IP-osoitealue.
dhcp-range=10.0.0.2,10.0.0.254,255.255.255.0,72h
```

Kuva 12. DHCP-palvelimen asetukset, dnsmasq.conf -tiedosto

Dnsmasq-ohjelmisto vaatii sen, että hakemistoon /var/lib/misc luodaan tiedosto dnsmasq.leases. Tähän tiedostoon dnsmasq tallentaa DHCP-palvelimeen yhteydessä olevan työaseman tiedot. Työaseman hakiessa verkkoasetuksia DHCP-palvelimelta ensimmäistä kertaa, dnsmasq tallentaa työaseman nimen ja laite- ja IP-osoitteet dnsmasq.leases -tiedostoon. Tietojen ollessa tallennettuna ja työaseman ottaessa seuraavan kerran yhteyttä DHCP-palvelimeen, palvelin osaa antaa viime yhteydessä olleet verkkoasetukset työasemalle. Työasema saa joka kerta samat verkkoasetukset.

Dnsmasq-sovelluksen ollessa käynnissä, kehitysalustan alapuolella oleva työasema saa verkkoasetuksensa alustan DHCP-palvelimelta, kuten kuvasta 13 näkyy. Ulkoverkkoon työasemalla ei saada vielä yhteyttä, koska reitittimen asetuksia ei ole vielä määritelty.

```
user@user-desktop:~$ ifconfig eth0
eth0      Link encap: Ethernet  HWaddr  00:0c:6e:7d:c6:e7
          inet addr:10.0.0.239  Bcast:10.0.0.255  Mask 255.255.255.0
          inet6addr: fe80::20c:6eff:fe7d:c6e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:848422 errors:0 dropped:0 overruns:0 frame:0
          TX packets:309228 errors:0 dropped:32 overruns:0 frame:0
          collisions:0 txqueuelen:1000
          RX bytes:877489066 (877.4 MB)  TX bytes:22022553 (22.0 MB)
          interrupt:19 Base address:0x9800
```

Kuva 13. Työaseman eth0-portin verkkotiedot

Iptables-ohjelmistolla määritellään mihin suuntaan reitittimen verkkoliikenteen kulku on sallittua. Työssä käytettiin yksinkertaista reititintä ilman palomuuria. Palomuuri otetaan käyttöön työn myöhemmässä vaiheessa. Reititin määriteltiin sallimaan kaikki liikenne ulkoverkosta sisäänpäin ja sisäverkosta ulospäin. Nämä määritykset on esitetty kuvassa 14.

```
# Tyhjennetään kaikki vanhat reititysasetukset.
iptables -F
# Sallitaan liikenne ulkoverkosta sisäverkkoon päin.
iptables -P INPUT ACCEPT
# Sallitaan liikennöinti verkosta toiseen verkkoon.
iptables -P FORWARD ACCEPT
# Sallitaan liikenne sisäverkosta ulkoverkkoon päin.
iptables -P OUTPUT ACCEPT
# Ohjataan liikennöinti verkosta 10.0.0.0 eteenpäin.
iptables -A FORWARD -i eth1 -s 10.0.0.0/255.255.255.0 -j ACCEPT
# Ohjataan ulkoverkosta tuleva liikenne verkkoon 10.0.0.0.
iptables -A FORWARD -i eth0 -d 10.0.0.0/255.255.255.0 -j ACCEPT
# Asetetaan NAT päälle.
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Kuva 14. Iptables-asetukset, sallitaan kaikki verkkoliikenne reitittimen läpi

Kuvassa 14 määritellään myös NAT (Network Address Translation) päälle. Tällä säästetään IPv4-osoitteita, koska jos kehitysalustan alapuolella on useampi työasema, niin NATin avulla kaikki kehitysalustan alapuolella olevat työasemat kommunikoivat ulkoverkkoon samalla IP-osoitteella. Reitittimen asetuksien ollessa kunnossa, reititys asetetaan päälle komennolla:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Komento sallii IP-osoitteiden ohjaamisen eteenpäin. Haettaessa verkkoasetukset uudelleen työasemalle huomataan, että työasema saa saman IP-osoitteen kuin aiemminkin ja työasemalla on yhteys ulkoverkkoon.

5.2 Siltaus

Toinen tietoliikenneominaisuus, joka on hyvä sisäistää, on kehitysalustan määrittäminen siltaavaksi laitteeksi. Kehitysalustaa määriteltäessä siltaavaksi laitteeksi, on käyttöjärjestelmässä oltava asennettuna bridge-utils -ohjelmisto. Bridge-utils

on ohjelmisto, jonka avulla on helppo luoda siltaavia yhteyksiä ethernet-porttien välille. Järjestelmässä bridge-utils sijaitsee hakemistossa /usr/sbin ja kantaa nimeä brctl. Ohjelmiston käyttämät komennot nähdään komennolla brctl. Kuvassa 15 on luotuna yksinkertainen siltaus eth0- ja eth1 porttien välille.

```
brctl addbr NGWsilta
brctl addif NGWsilta eth0
brctl addif NGWsilta eth1
ifconfig NGWsilta up
```

Kuva 15. Kehitysalustan määritteleminen siltaavaksi laitteeksi bridge-utils -sovelluksella

Kuvassa 15 luodaan uusi siltaus, jonka nimenä on NGWsilta. Nimellä kuvataan siltausta, koska siltaus on rakennettu NGW100-kehitysalustan ylitse. Siltaukseen on lisätty kehitysalustan eth0- ja eth1-portit, mikä tarkoittaa sitä, että verkkoliikenne sillataan näiden porttien yli. Lopuksi siltaus asetetaan päälle. Kehitysalustan alapuolella olevalle työasemalle haetaan verkkoasetukset uudelleen, jolloin työasema saa IP-osoiteen ulkoverkosta. Tässä tapauksessa ulkoverkolla tarkoitetaan Vaasan ammattikorkeakoulun verkkoa.

```
user@user-desktop:~$ ifconfig eth0
eth0      Link encap: Ethernet  HWaddr 00:0c:6e:7d:c6:e7
          inet addr:192.168.69.7  Bcast:192.168.69.255  Mask 255.255.255.0
          inet6addr: fe80::20c:6eff:fe7d:c6e7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:848464 errors:0 dropped:0 overruns:0 frame:0
          TX packets:309254 errors:0 dropped:32 overruns:0 frame:0
          collisions:0 txqueuelen:1000
          RX bytes:877493671 (877.4 MB)  TX bytes:22026889 (22.0 MB)
          interrupt:19 Base address:0x9800
```

Kuva 16. Työaseman eth0-portin verkkotiedot

Kuvasta 16 huomataan työaseman verkkoasetuksien tulevan ulkoverkosta, eli Vaasan ammattikorkeakoulun DHCP-palvelimelta. Kuitenkin jos kehitysalustaa ei haluta enää käyttää siltaavana laitteena, voidaan siltaus asettaa pois päältä komennolla:

```
ifconfig NGWsilta down
```

6 SULAUTETTU WEB-PALVELIN

6.1 Web-palvelin

Web-palvelin on laite, joka pitää sisällään erilaisia tiedostoja, joita se jakaa asiakkailleen käyttäen HTTP-protokollaa (Hypertext Transfer Protocol). Yleensä nämä tiedostot ovat html-sivuja, kuvia, videoita, ääniä tai ohjelmia. Näihin tiedostoihin asiakkaat voivat olla yhteydessä selainohjelmillaan, joita ovat mm. Mozilla Firefox, Google Chrome ja Internet Explorer. Asiakkaiden ottaessa yhteyttä web-palvelimeen selaimella, selain lähettää palvelimelle HTTP-pyynnön ja palvelin palauttaa selaimelle HTTP-vastauksena jonkin tiedoston.

Työssä sulautettuna web-palvelimena käytetään NGW100-kehitysalustaa, jonka web-palvelimena toimii lighttpd-ohjemisto (lighty). Ohjelmistoa kuvaavat erityisesti adjektiivit kevyt, nopea, turvallinen ja joustava. Ohjelmisto valittiin käytettäväksi työhön sen keveyden takia. Lighttpd-ohjelmistolla saadaan luotua kevyt web-palvelin, joka käyttää vähän järjestelmämuistia, verrattuna muihin web-palvelinohjelmistoihin. Työssä ei perehdytä Lighttpdn tietoturvaominaisuuksiin, koska web-palvelimen tietoturva toteutetaan työn myöhemmässä vaiheessa palomuurilla ja VPN-yhteydellä. /8/

6.2 Web-palvelimen pystyttäminen

Yksinkertaisen web-palvelimen pystyttäminen käy nopeasti lighttpd-ohjelmistolla. Pystyttämiseen vaaditaan määrittelytiedosto lighttpd.conf ja lokitiedostot access.log ja error.log. Nämä tiedostot sijoitetaan selkeyden kannalta /etc/lighttpd -hakemistoon. Tiedostolla lighttpd.conf määritellään se miten web-palvelin toimii. Kuvassa 17 on esiteltynä työssä käytetyt määrittelyt. Lokitiedostot access.log ja error.log, pitävät yllä lokia siitä, minkälaisia yhteyksiä palvelimeen on muodostettu ja mitä virheitä yhteyksissä on tapahtunut. Työssä web-palvelimelle luotiin myös html-tiedosto, joka avautuu, kun palvelimeen muodostetaan yhteys.

```
# Web-palvelin juurihakemisto.
server.document-root = "/etc/lighttpd"
# Mitä http-porttia palvelin käyttää.
server.port = 80
# Palvelimen IP-osoite tai hostname.
server.bind = "10.0.0.1"

# Tiedoston error.log sijainti.
server.errorlog = "/etc/lighttpd/error.log"
# Tiedoston access.log sijainti.
accesslog.filename = "/etc/lighttpd/access.log"

# Web-palvelimen tukemat moduulit.
server.modules = (
# Moduuli, jolla web-palvelin kirjoittaa accesslokia.
"mod_accesslog"
)

# Tiedostomuodot, joita web-palvelin tukee.
mimetype.assign = (
".html" => "text/html",
)
# Html-sivusto, joka asiakkaalle avataan.
index-file.names = ("html-tiedosto.html")
```

Kuva 17. Sulautetun web-palvelimen määritykset, lighttpd.conf -tiedosto

Web-palvelimelle muodostettaessa yhteys asiakkaan selainohjelmalla, palvelin palauttaa selaimelle html-tiedoston. Tätä ennen lighttd-ohjelmisto on kuitenkin oltava käynnissä. Lighttp saadaan käynnistettyä kuvassa 17 näkyvällä määrittely-tiedostolla, seuraavalla komennolla:

```
lighttpd -f lighttpd.conf
```

7 TIETOTURVALLINEN YHTEYS

7.1 OpenSSH

OpenSSH (OpenBSD Secure Shell) on ilmainen salatuissa tietoliikenneyhteyksissä käytettävä ohjelmisto, johon internetin teknilliset käyttäjät luottavat. OpenSSH salaa kaiken SSH-yhteydessä kulkevan liikenteen, mukaan lukien salasanat. Yhtään viestiä ei siis lähetetä selkokielisenä. Jotkut etäyhteystyökalut, kuten telnet, rlogin ja ftp eivät tätä tee. Näiden työkalujen käyttäjät eivät välttämättä havainnollista sitä, että heidän liikennöintinsä suoritetaan Internetin ylitse selkokielisenä, eli ilman salausta. Tällöin salasanatkin kulkevat Internetin ylitse selkokielisinä. /12/

OpenSSH pitää sisällään todentamisen. Todentamisella varmistetaan se, että SSH-palvelimeen yhteydessä oleva asiakas on todella se, kuka se väittää olevansa. Todentaminen suoritetaan julkisesta ja yksityisestä avaimesta muodostetulla avainparilla. Näistä yksityinen avain suojataan lisäksi vielä salasanalla. OpenSSH pitää sisällään monta ohjelmaa, joiden avulla todentamista käyttävä SSH-yhteys rakennetaan. Näistä ohjelmista on hyvä ymmärtää seuraavien ohjelmien tarkoitus: /12/

- sshd
  SSH-palvelinohjelma, jota SSH-palvelin käyttää. Palvelin odottaa asiakkaan yhteydenottoa. Asiakkaan ottaessa yhteyttä palvelimeen, palvelin suorittaa todentamisen, jonka jälkeen asiakasta ryhdytään palvelemaan. Tämän ohjelman toimivuus määritellään tiedostolla sshd_config.
- ssh
  SSH-asiakasohjelma, jota SSH-yhteyden asiakas käyttää muodostaessaan yhteyden palvelimeen. Tämän ohjelman toimivuus määritellään tiedostolla ssh_config.
- ssh-keygen
  Ohjelmisto, jolla luodaan todentamisessa tarvittavat komponentit. Ohjelmistolla luodaan julkinen avain, yksityinen avain ja yksityistä avainta suojaava salasana.

- scp ja ssh-copy-id
  Ohjelmistot, joilla pystytään turvallisesti kopioimaan tiedostoja asiakkaalta palvelimelle. Tätä ohjelmistoa tulee käyttää mm. julkisen avaimen siirrossa.

7.1.1 SSH-yhteys

Työssä SSH-yhteys muodostetaan palvelimena toimivan NGW100-kehitysalustan ja Linux Ubuntu -työaseman välille. Palvelimelle on aikaisemmin asennettu tarvittava openssh-ohjelmisto, mutta tässä vaiheessa asiakkaalle on asennettava openssh-client -ohjelmisto. Sovelluksen asentaminen suoritetaan komennolla:

```
sudo apt-get install openssh-client
```

SSH-yhteys muodostetaan aluksi yksinkertaisella kytkennällä, jolloin yhteyden osapuolet sijaitsevat samassa verkossa. Kytkentä toteutetaan niin, että kehitysalustan eth1-portti (LAN) kytketään työasemalle. Työasemalle määritellään kiinteät verkkoasetukset ja IP-osoitteeksi 10.0.0.2.

Palvelimena toimivaa kehitysalustaa on tähän mennessä käytetty pelkällä pääkäyttäjällä, mutta perehdyttäessä tietoturvallisuuteen, kehitysalustalle luodaan ns. normaali käyttäjä. Käyttäjä luodaan komennolla:

```
adduser -h user user
```

Tällä komennolla järjestelmään luodaan uusi käyttäjä ja käyttäjälle salasana ja kotihakemisto. Käyttäjän kotihakemistoon luodaan alihakemisto .ssh, jonka sisälle luodaan julkista avainta sisällä pitävä tiedosto. Tiedoston nimeksi asetetaan authorized_keys. Tästä eteenpäin kehitysalustaa käytetään normaalilla käyttäjällä. Tämä käyttäjä saadaan käyttöön komennolla:

```
su user
```

Perusasioiden ollessa kunnossa, SSH-yhteyttä testataan muodostamalla SSH-yhteys asiakkaalta palvelimelle. Tässä vaiheessa ei vielä käytetä julkista ja

yksityistä avainta, vaan palvelimelle kirjaudutaan sisään käyttäjätunnuksella. Palvelimella voidaan käyttää aikaisemmin asetettuja verkkoasetuksia, joten tällöin eth1-portti, johon SSH-yhteys muodostetaan, käyttää IP-osoitetta 10.0.0.1. SSH-yhteys muodostetaan työaseman komentoriviltä komennolla:

```
ssh user@10.0.0.1
```

7.1.2 Todentaminen julkisella ja yksityisellä avaimella

Todentaminen perustuu avainpariin, joka muodostetaan julkisesta ja yksityisestä avaimesta. Asiakkaan tulee omistaa omasta avainparistaan yksityinen avain ja palvelimen tulee omistaa asiakkaan yksityistä avainta vastaava julkinen avain. SSH-yhteydessä yleisimmin käytettyjä avainparimahdollisuuksia ovat RSA ja DSA. Näiden mahdollisuuksien väliltä ei löytynyt merkittävää tietoturvaeroa, joten työn toteuttamisen kannalta ei ole väliä kumpaa käytetään. Todentamista testattiin kummallakin mahdollisuudella ja molemmat toimivat moitteettomasti. Työssä esitellään RSA-avainparin luonti. RSA-avainparin luonti suoritetaan komennolla:

```
ssh-keygen -t rsa
```

Tätä komentoa suoritettaessa ssh-keygen kysyy, halutaanko yksityinen avain suojata salasanalla. Yksityinen avain on syytä suojata salasanalla, koska jos yksityinen avain sattuu joutumaan vääriin käsiin, kuka tahansa voi käyttää sitä. Komentoa suoritettaessa on huomattava kenelle työaseman käyttäjälle avainpari luodaan. Käyttäjä, kenellä tämä komento suoritetaan, luodaan myös avainpari, eivätkä muut käyttäjät voi käyttää samaa avainparia. Komento luo julkisen avaimen id_rsa.pud ja yksityisen avaimen id_rsa. Nämä avaimet sijoitetaan automaattisesti käyttäjän kotihakemistossa olevaan alihakemistoon .ssh. Tästä avainparista palvelimen tulee omistaa asiakkaan julkinen avain todentamista varten. Julkinen avain kopioidaan palvelimelle komennolla:

```
scp /home/user/.ssh/id_rsa.pub user@10.0.0.1:/user/.ssh/authorized_keys
```

Komento scp on lyhenne sanoista secure copy, eli turvallinen kopiointi. Komento kopioi julkisen avaimen käyttäen SSH-yhteyttä, joten liikenne on salattu ja tällöin ei-halutut osapuolet eivät pääse käsiksi siihen. Julkinen avain kopioidaan normaalin käyttäjän kotihakemiston .ssh-alihakemistossa olevaan authorized_keys -tiedostoon. Secure copyn sijasta voidaan julkisen avaimen kopiointiin käyttää myös ohjelmistoa ssh-copy-id. Tätä ohjelmistoa käytettäessä kehitysalustan normaalilla käyttäjällä ei tarvitse olla valmiiksi luotuna alihakemistoa .ssh ja tiedostoa authorized_keys, vaan ohjelmisto luo hakemiston ja tiedoston itsestään normaalin käyttäjän kotihakemistoon.

Palvelimelle on kerrottava, että se etsii julkista avainta normaalin käyttäjän kotihakemistosta, koska oletusasetuksilla sshd etsii julkista avainta pääkäyttäjän kotihakemistosta. Kehitysalustaa on siirryttävä käyttämään pääkäyttäjällä, koska normaalilla käyttäjällä ei ole oikeuksia määrittelytiedostoihin. Pääkäyttäjällä muokataan SSH-palvelimen määrittelytiedostoa sshd_config. Tiedostosta on otettava käyttöön AuthorizedKeysFile -rivi ja siihen on lisättävä polku julkiseen avaimeen. Rivi tulee näyttää seuraavanlaiselta:

```
AuthorizedKeysFile /user/.ssh/authorized_keys
```

Julkinen avain voidaan nyt poistaa työasemalta, koska työaseman ei tarvitse omistaa itsellään omaa julkista avaintaan. Tämä tehdään turvallisuus syistä, koska jos ei-haluttu osapuoli pääse käsiksi työasemaan, hän ei saa varastettua avainparin molempia osapuolia.

Asiakastyöasemalta voidaan nyt ottaa SSH-yhteys palvelimelle samalla komennolla kuin aikaisemminkin. Yhteyden muodostusvaiheessa, asiakas liittää yksityisen avaimensa viestiinsä ja palvelin tarkistaa, että tämä yksityinen avain sopii yhteen julkisen avaimen kanssa. Nyt kuitenkin huomataan, että ennen kuin SSH-yhteys avataan, ssh-ohjelma vaatii syötettäväksi salasanan, joka suojaa yksityistä avainta. Salasanan ollessa oikea ja avaimien sopiessa yhteen, SSH-yhteys avataan. Tapauksissa, joissa salasana tai avaimet eivät täsmää, yhteyden muodostaminen torjutaan.

7.1.3 SSH-etäyhteys

Kehitysalustalla toimivalle palvelimelle muodostetaan SSH-etäyhteys. Etäyhteydessä asiakas ja palvelin sijaitsevat eri verkoissa. Työssä kehitysalustan eth0-portti kytketään julkiseen verkkoon ja työasema kytketään Vaasan ammattikorkeakoulun verkkoon. Kehitysalustan ollessa kytkettynä ulkoverkkoon, siihen lisätään palomuuri iptables-työkalulla. Palomuuri määritellään estämään kaikki muut yhteydenottopyynnöt, paisti SSH-pyynnöt. Palomuuriin tehdään lisää aukkoja työn myöhemmässä vaiheessa. Tässä vaiheessa muodostettu palomuuri on pystytetty kuvassa 19.

```
# Tyhjennetään kaikki vanhat iptables asetukset.
iptables -F
# Estetään palvelimelle julkisesta verkosta tuleva liikenne.
iptables -P INPUT DROP
# Sallitaan julkisesta verkosta vain SSH-yhteyden muodostaminen palvelimelle.
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

Kuva 19. Estetään julkisesta verkosta tulevat palvelupyynnöt, paitsi SSH-pyynnöt

SSH-etäyhteyden muodostaminen käy samalla tavalla kuin samassa verkossa olevien laitteiden välinen yhteys. Kehitysalusta on nyt kytkettynä julkiseen verkkoon, joten IP-osoite on vaihtunut. Etäyhteys muodostetaan komennolla:

```
ssh user@ip-osoite
```

7.2 OpenVPN

OpenVPN on OpenSSH:n tavoin ilmainen salatuissa tietoliikenneyhteyksissä käytettävä ohjelmisto, joka käyttää liikenteen salaamiseen SSL/TLS -protokollaa. VPN tulee sanoista Virtual Private Network, joka tarkoittaa virtuaalista yksityistä verkkoa. Tarkemmin tällä tarkoitetaan sitä, että kahden osapuolen väliseen liikennöintiin ei käytetä fyysisiä verkkokortteja, vaan molemmille laitteille luodaan virtuaaliset verkkokortit, joiden kautta liikennöinti suoritetaan.

VPN-yhteys suojataan julkisesta ja yksityisestä avaimesta muodostetulla avainparilla. Todentaminen toteutetaan lähes samalla periaatteella kuin SSH:ssa, mutta todentaminen suoritetaan kaksisuuntaisesti. SSH:ssa todentaminen suoritettiin yksisuuntaisesti, eli vain palvelin todentaa asiakkaan luotettavuuden, mutta VPN:ssä tämän lisäksi myös asiakas todentaa palvelimen luotettavuuden. Työssä muodostetussa VPN-yhteydessä käytetään seuraavia avaimia ja sertifikaatteja: /11/

- ca.crt
  Julkinen pääsertifikaatti, jolla allekirjoitetaan kaikki avaimet.
- server.crt
  Palvelimen julkinen avain, joka ei ole salainen.
- server.key
  Palvelimen yksityinen avain, joka on salainen.
- client.crt
  Asiakkaan julkinen avain, joka ei ole salainen.
- client.key
  Asiakkaan yksityinen avain, joka on salainen.
- ta.key
  TLS-salausavain (Transport Layer Security). Lisäsuojaus, jolla lähetettävät viestit allekirjoitetaan.
- dh1024.pem
  Diffie Hellman -muuttuja.

Avaimista ja sertifikaateista .crt-päätteiset tiedostot ovat julkisia ja .key-päätteiset yksityisiä. OpenVPN-ohjelmisto käyttää avaimia ja sertifikaatteja seuraavalla tavalla: VPN-yhteyttä muodostettaessa asiakas lähettää viestin, että haluaa muodostaa yhteyden palvelimen kanssa. Palvelin vastaanottaa viestin ja vastaa asiakkaalle. Vastaukseen palvelin liittää oman julkisen avaimensa (server.crt), joka on allekirjoitettu julkisella pääsertifikaatilla (ca.crt). Asiakas vastaanottaa viestin, josta asiakas poimii palvelimen julkisen avaimen itselleen. Seuraavaan viestiin, jonka asiakas lähettää, on liitetty asiakkaan julkinen avain (client.crt) allekirjoitettuna pääsertifikaatilla. Viesti on myös allekirjoitettu TLS-salausavaimella (ta.key) ja salattu palvelimen julkisella avaimella. Palvelimen vastaanottaessa viestin, se

purkaa salauksen omalla yksityisellä avaimellaan (server.key) ja tarkistaa TLS-salausavaimella, että viesti ei ole muuttunut matkalla. Viestit, joita ei ole varustettu salausavaimella, torjutaan ilman sen kummempia toimenpiteitä. Salausavaimella torjutaan mm. palvelunestohyökkäyksiä. Asiakkaan viestistä palvelin poimii myös asiakkaan julkisen avaimen, jolla palvelin salaa seuraavan viestinsä. Yksityiset avaimet ja salausavain on pidettävä salassa. /11/

7.2.1 VPN-etäyhteys

VPN-yhteys muodostetaan SSH:n tavoin palvelimena toimivan NGW100-kehitysalustan ja Ubuntu-työaseman välille. Etäyhteydessä palvelin ja asiakas sijaitsevat SSH-etäyhteyden tavoin eri verkoissa, asiakas Vaasan ammattikorkeakoulun verkossa ja palvelin julkisessa verkossa. VPN-etäyhteytenä käytetään tunneloivaa ratkaisua. Tunneloivalla ratkaisulla tarkoitetaan sitä, kun asiakas ottaa yhteyden palvelimeen, osapuolet muodostavat tunnelin, jonka liikenteeseen ei pääse käsiksi tunnelin ulkopuoliset osapuolet. Tunnelin pystyy muodostamaan vain luotettavat osapuolet. Tunnelilla tarkoitetaan virtuaalista verkkoa, joka toimii fyysisen verkon päällä. Osapuolien luotettavuus tarkistetaan kaksisuuntaisella todentamisella.

VPN-yhteydessä käytettävään todentamiseen tarvittavat avaimet luodaan työasemalla, koska sen käyttöjärjestelmä on joustavampi tähän tarkoitukseen. Ennen avaimien luontia, työasemalle asennetaan tarvittava openvpn-ohjelmisto komennolla:

```
sudo apt-get install openvpn
```

Sovelluksen asentamisen jälkeen, yhteydessä tarvittavien avaimien luontitiedostot asettuvat hakemistoon /usr/share/doc/openvpn/examples/easy-rsa/2.0. Tämän hakemiston sisältö kopioidaan työasemalla haluttuun paikkaan. Työssä hakemiston sisältö kopioitiin hakemistoon /etc/openvpn/easy-rsa. Komentorivillä siirrytään easy-rsa -hakemistoon ja siellä luodaan todentamiseen vaadittavat avaimet ja sertifikaatit. Työssä käytetyt avaimet ja sertifikaatit on luotuna kuvassa 20. /11/

```
# Siirrytään pääkäyttäjäksi.
sudo su
# Siirrytään easy-rsa -hakemistoon
cd /etc/openvpn/easy-rsa
# Valitaan kohteeksi vars-tiedosto.
source vars
# Tyhjennetään ja poistetaan kaikki avaimet.
./clean-all
# Luodaan pääsertifikaatti.
./build-ca
# Luodaan julkinen ja yksityinen avain palvelimelle.
./build-key-server server
# Luodaan yksityinen avain ja yksityistä avainta suojaava salasana asiakkaalle
./build-key-pass client
# Luodaan Diffie Hellman muuttuja.
./build-dh
# Siirrytään easy-rsa -hakemiston alihakemistoon keys
cd keys
# Luodaan TLS-salausavain
openvpn --genkey --secret ta.key
```

Kuva 20. Tarvittavien avainten ja sertifikaattien luonti /11/

Avaimet on nyt luotuna hakemistoon /etc/openvpn/keys. Avaimista palvelimelle on siirrettävä avaimet ca.crt, server.crt, server.key, ta.key ja dh1024.pem. Nämä avaimet sijoitetaan palvelimelle /etc/openvpn -hakemistoon. Asiakkaalla on oltava avaimet ca.crt, client.crt, client.key ja ta.key, jotka sijoitetaan työaseman hakemistoon /etc/openvpn.

VPN-osapuolien toiminnat määritellään tiedostoissa server.conf ja client.conf. Palvelimella on tärkeää määritellä se, että mikään liikenne ei kulje muuta kuin sen virtuaalisen verkkokortin kautta. Työssä käytetty palvelimen määrittelytiedosto on esitelty kuvassa 21 ja asiakkaan kuvassa 22.

```
# Mitä TCP/UDP -porttia VPN-palvelin kuuntelee.
port 1194
# TCP- vai UDP-palvelin.
proto udp
# Käytetään VPN-tunnelia.
dev tun
# Mitä avaimia ja sertifikaatteja palvelin käyttää.
ca ca.crt
cert server.crt
key server.key
# Diffie Hellman -muuttuja.
dh dh1024.pem
# Määritellään VPN-aliverkko.
server 10.0.1.0 255.255.255.0
# Tiedosto, joka ylläpitää asiakkaiden IP-osoitteita.
ifconfig-pool-persist ipp.txt
# Kaikki liikenne ohjataan kulkemaan virtuaalisen verkkokortin kautta
push ”redirect-gateway def1”
# Lähetetään asiakkaalle viesti joka kymmenen sekunnin välein ja viestiin pitää
# tulla vastaus 120 sekunnin sisällä.
keepalive 10 120
# Käytetään TLS-salausavainta.
tls-auth ta.key 0
# Tiedosto, jossa pidetään listaa VPN-yhteyden osapuolista. Lista tyhjennetään ja
# päivitetään minuutin välein.
status openvpn-status.log
```

Kuva 21. Palvelimen määrittelytiedosto server.conf

```
# Tämä osapuoli on asiakas.
client
# käytetään VPN-tunnelia. Pitää olla sama kuin palvelimella.
dev tun
# Yhteys otetaan UDP-palvelimeen.
proto udp
# Palvelimen IP-osoite ja portti
remote ip-osoite 1194
# Ei sidota asiakasta käyttämään vain tiettyä porttia.
nobind
# Mitä avaimia asiakas käyttää.
ca ca.crt
cert client.crt
key client.key
# Käytetään TLS-salausavainta.
tls-auth ta.key 1
```

Kuva 22. Asiakkaan määrittelytiedosto client.conf

VPN-yhteyden osapuolet ovat nyt valmiita käynnistettäviksi. Tätä ennen palomuuriin tehdään kuitenkin aukko, jotta VPN-yhteyden muodostaminen palvelimelle sallitaan. Palomuuriin tehdä aukko, joka sallii sisään tulevan UDP-liikenteen portin 1194 kautta. Tämä suoritetaan komennolla:

```
iptables -A INPUT -p udp --dport 1194 -j ACCEPT
```

VPN-yhteys voidaan nyt käynnistää. VPN-yhteydet käynnistetään määrittelytiedostoilla komennoilla:

```
# Palvelin
openvpn --config server.conf
# Asiakas
openvpn --config client.conf
```

Komentoja syötettäessä on kuitenkin huomioitava se, että komentorivillä ollaan oikean hakemiston sisällä. Komentojen mentyä läpi, VPN-yhteys on muodostettuna palvelimen ja asiakkaan välille. Asiakaan verkkoasetuksia tarkasteltaessa, huomataan että verkkoasetuksiin on ilmestynyt ylimääräinen virtuaalinen verkkokortti tun0, joka on saanut verkkoasetuksena palvelimen virtuaaliselta verkkokortilta. Tämä näkyy kuvassa 23.

```
user@user-desktop:~$ ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr  00-00-00-00-00-00-00-00-00-00-00-
          00-00-00-00
          inet addr:10.0.1.6  P-t-P:10.0.1.5  Mask 255.255.255.255
          UP POINTTOPOINT RUNNING NOARP MULTICAST  MTU:1500
          Metric:1
          RX packets:3 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:100
          RX bytes:252 (252.0 B)  TX bytes:252 (252.0 B)
```

Kuva 23. Asiakkaan tun0-portin verkkoasetukset

VPN-yhteyden muodostamisen jälkeen palomuuriin tehdään yksi aukko lisää. Tämän aukon kautta sulautetulle alustalle sallitaan kaikki sisään tuleva verkkoliikenne virtuaalisen tun0-portin kautta. Tällä mahdollistetaan se, että VPN-yhteydellä saadaan avattua sulautetun web-palvelimen ylläpitämä html-sivusto. Aukko tehdään komennolla:

```
iptables -A INPUT -i tun0 -j ACCEPT
```

Palomuurin salliessa kaiken sisään tulevan liikenteen tun0-portin kautta, web-palvelimen ylläpitämälle html-sivustolle saadaan yhteys asiakkaan selainohjelmalla. Lopputuloksena on se, että palvelimelle ei saada muodostettua muita kuin tietoturvallisia yhteyksiä, joita ovat VPN-yhteys ja aikaisemmin pystytetty SSH-yhteys.

8 YHTEENVETO

Työn pääasiallinen tarkoitus oli tutustua sulautetun web-palvelimen tietoturvaan. Sulautetun web-palvelimen kehitysalustana toiminut Atmel Mature NGW100 Network Gateway Kit soveltui mainiosti työn toteuttamiseen sen hyvien liitäntöjen ja verkko-ominaisuuksien ansiosta. Kehitysalustan liitännöistä jatkuvassa käytössä olivat Ethernet-portit, MMC/SD-korttipaikka ja RS232-sarjaportti.

NGW100-kehitysalustan liitäntöihin ja toimintaan tutustumien sujui ongelmitta ja kehitysalustaan päästiin hyvin sisälle. Tämän jälkeen kehitysalustalla ryhdyttiin luomaan sulautettua Linux-käyttöjärjestelmää, joka osoittautui työn haastavimmaksi osuudeksi. Sulautetun Linux-järjestelmän rakentamiseen kului todella paljon turhaa aikaa, koska Buildroot tuotti paljon ongelmia. Buildrootin patch-tiedostot pitivät sisällään vanhoja polkuja asennettaviin sovelluksiin, eikä käyttöjärjestelmä kääntynyt loppuun saakka. Muitakin toiminnallisia ongelmia Buildrootissa ilmeni, mutta näitä ei kuitenkaan saatu ratkottua. Lopuksi Buildrootista ilmestyi uusi versio, buildroot-2011.05. Tällä versiolla käyttöjärjestelmän rakentaminen onnistui vaivatta, jonka jälkeen ongelmatkin vähenivät ja työ lähti edistymään hyvää vauhtia.

Sulautetun web-palvelin tietoturvaan perehtymisen ohella työssä oli syvennyttävä kehitysalustan tietoliikennemahdollisuuksiin. Tässä vaiheessa ongelmia tuotti ohjeiden soveltaminen. Useimmat ohjeet, joita työssä käytettiin, olivat tehty Ubuntulle tai muille Linux-järjestelmille, joten näitä ohjeita oli sovellettava sulautetulle Linux-järjestelmälle. Sama ongelma toistui myös työn muissa osioissa. Näistä ongelmista selvittiin ja kehitysalusta saatiin toimimaan reitittimenä ja siltaavana laitteena.

Tietoturvaa päästiin käsittelemään, kun kehitysalustan tietoliikenteeseen ja näin ollen Ethernet-porttien toimintaan oli perehdytty tarkemmin. SSH-yhteyden muodostaminen tuotti pienin ongelman, koska SSH-palvelimena käytettiin aluksi dropbear-ohjelmistoa. Dropbear-sovelluksella todentaminen julkisella ja yksityisellä avainparilla ei kuitenkaan sopinut yhteen OpenSSH asiakaan kanssa. Tätäkään ongelmaa ei saatu ratkaistua muulla tavalla kuin luopumalla dropbearista ja

tilalle oli asennettava OpenSSH. Tämän jälkeen todentaminen ja näin ollen SSH-yhteyden luominen onnistui.

VPN-yhteyden muodostamisessa ei juuri ollut ongelmia, koska OpenVPN-ohjelmiston omalta internet-sivustolta löytyi suoraviivaiset ohjeet VPN-yhteyden pystyttämiseen. VPN-yhteydessä käytettävien avaimien ja sertifikaattien ymmärtäminen tuotti kuitenkin hieman päänvaivaa.

Kaikin puolin työ oli opettavainen ja piti sisällän paljon tietotekniikan koulutusohjelmaan sisältyneiden kurssien sisältöä. Esimerkiksi VPN-yhteyden pystyttäminen asiakkaan ja palvelimen välille oli aikaisemmin suoritettu tietoliikenteen kurssilla. Työtä suorittaessa suurin oppiminen tapahtui Linuxin käytössä, koska kehitysalustan sulautettu käyttöjärjestelmä on käytettävissä vain Linux-komentojen avulla ilman minkäänlaista graafista käyttöliittymää. Linuxiin perehtyminen oli palkitsevaa, koska ennen työn suorittamista, omat Linux-taitoni eivät olleet kovinkaan hyvät.

LÄHDELUETTELO

/1/ Atmel Corporation, *Mature NGW100 Network Gateway Kit*. Viitattu 2.11.2011. http://www.atmel.com/dyn/products/tools_card.asp?catecory_id=163&family_id=607&subfamily_id=2138&tool_id=4102

/2/ AVR Freaks Wiki*, Documentation:NGW/NewKernel, Installing the new kernel and filesystem on the SD Card.* Viitattu 9.11.2011. http://www.avrfreaks.net/wiki/index.php/Ducumentation:NGW/NewKernel

/3/ AVR Freaks Wiki*, Documentation:NGW/NewKernel, Using Buildroot to create a new kernel*. Viitattu 9.11.2011. http://www.avrfreaks.net/wiki/index.php/Documentation:NGW/NewKernel

/4/ AVR Freaks Wiki, *NGW/FirstSteps, USB*. Viitattu 2.11.2011. http://www avrfreaks.net/wiki/index.php/Documentation:NGW/FirstSteps

/5/ AVR Freaks Wiki, *NGW/NGW100 Hardware* reference. Viitattu 2.11.2011. http://www.avrfreaks.net/wiki/index.php/Documentation:NGW/NGW100_Hardware_reference

/6/ AVR Freaks Wiki*, NGW/Quick Start, Using the NGW as a router*. Viitattu 2.11.2011. http://www.avrfreaks.net/wiki/index.php/Documentation:NGW/Quick_Start

/7/ Buildroot, *About Buildroot*. Viitattu 2.11.2011. http://buildroot.uclib.org/downloads/buildroot.html

/8/ Lighttpd. Viitattu 9.11.2011. http://www.lighttpd.net

/9/ Linux.fi, *Palomuuri*. Viitattu 14.12.2011. http://linux.fi/wiki/Palomuuri

/10/ Linux.fi, *Tietoturva*. Viitattu 14.11.2011. http://linux.fi/idex.php/Tietoturva

/11/ Open Source VPN, *HOWTO, Setting up your own Certificate Authority (CA) and generating certificates and keys for an OpenVPN server and multiple clients*. Viitattu 22.11.2011. http://openvpn.net/index.php/open source/documentation/howto.html

/12/ OpenSSH, *OpenSSH FAQ (Frequently asked questions), 1.0 - What Is OpenSSH and Where I Can Get It?*. Viitattu 7.12.2011. http://www.openssh.com/faq.html

/13/ SecMeter, *Palvelunestohyökkäys*. Viitattu 14.12.2011. http://www.secmeter.com/palvelunestohyokkays.html

/14/ Tietoturvaopas, *Haittaohjelmat*. Viitattu 14.12.2011. http://www.tietoturvaopas.fi/uhatjaniidentorjunta/haittaohjelmat.html

SULAUTETTU VERKON YHDYSKÄYTÄVÄ

Ver. 0.9 Jussi Haapamäki 20.1.2012

Tämän harjoitustyön tarkoituksena on tutkia sulautetun kehitysalustan perustieto-liikenne-mahdollisuuksia. Sulautettuna kehitysalustana toimii Atmel Mature NGW100 Network Gateway Kit, jota käytetään sulautetulla Linux-käyttöjärjestelmällä. Työssä tutustutaan sulautetun Linux-käyttöjärjestelmän käyttöönottoon, NGW100-kehitysalustan tietoliikenteeseen, sulautettuun web-palvelimeen ja SSH-yhteyteen. Tietoliikenne osuus pitää sisällään reitityksen, DHCP-palvelimen ja siltauksen. SSH-osuus pitää sisällään SSH-yhteyden luomisen ja todentamisen yksityisestä ja julkisesta avaimesta muodostetulla avainparilla ja SSH-tunneloinnin.

Esitehtävät:

Tehtävä 1.

Katso työohjeista kohta: Osa 1, Tehtävä 1, kohta viisi. Mitä fdisk-työkalun komentoja tässä kohdassa tulee käyttää. Ohjeita löytyy osoitteesta: http://linux.fi/wiki/Fdisk.

Tehtävä 2.

Kuinka määrittelet Linuxissa verkkoasetukset käyttäen interfaces-tiedostoa. Verkkoasetukset määritellään porteille eth0 ja eth1. Asetuksissa eth0-portin tulee saada verkkoasetuksensa DHCP-palvelimelta ja eth1-portin verkkoasetukset tulee olla kiinteät. Kiinteissä verkkoasetuksissa määritellään IP-osoite, verkko-osoite, maski ja broadcast-osoite. Ohjeita löytyy osoitteesta: http://linux.fi/wiki/Verkkoliitynnät.

Tehtävä 3.

Kuinka käytät bridge-utils -työkalulla komentoja addbr ja addif, luodaksesi yksinkertaisen siltauksen eth0- ja eth1-porttien ylitse. Yksinkertaisen siltauksen luontiin tarvitaan kolme komentoa.

Osa 1. Sulautettu Linux-käyttöjärjestelmä

Tehtävä 1. SD-kortin alustus, osiointi ja ext2-tiedostojärjestelmän luonti

1. Kytketään SD-kortti työasemalle kortinlukijaan ja avataan komentorivi (Terminal).
2. Siirrytään komentoriviltä root-käyttäjäksi.

```
sudo su
```

3. Irrotetaan SD-kortin tiedostojärjestelmä työasemalta.

```
umount /dev/sdd1
```

SD-kortin laitetunnus voi olla myös jokin muukin kuin sdd, esimerkiksi sda, sdb, sdc, sde.

4. Siirrytään SD-kortin osionti tilaan.

```
fdisk /dev/sdd
```

5. SD-kortti alustetaan ja osioidaan. Käytä esitehtävässä yksi selvitettyjä komentoja.

```
# Syötetään komento, jolla tarkastellaan mitä osoita SD-kortilla on

# Poistetaan kortilla oleva osio

# Lisätään uusi osio

# Valitaan osioksi ensiöosio (primary partition)

# Valitaan osion numeroksi yksi

# Osiolle valitaan käyttöön SD-kortin koko kapasiteetti

# Kirjoitetaan asetukset kortille
```

6. Luodaan SD-kortille ext2-tiedostojärjestelmä.

```
mke2fs /dev/sdd1
```

Tehtävä 2. Sulautetun Linux-käyttöjärjestelmän siirto SD-kortille

1. Pysytään komentorivillä edelleen root-käyttäjänä ja luodaan työaseman työpöydälle uusi hakemisto, jonka nimeksi asetetaan sd-kortti.
2. Liitetään SD-kortti yhteen sd-kortti-hakemiston kanssa.

```
mount /dev/sdd1 /home/user/Desktop/sd-kortti
```

Tarkista mitä että SD-kortille on syntynyt hakemisto lost&found.

3. Siirrytään komentorivillä sd-kortti -hakemistoon ja puretaan sinne sulautetun Linux-käyttöjärjestelmän juuritiedostojärjestelmä.

```
tar -xf /home/user/Desktop/buildroot-2011.05/output/images/rootfs.tar
```

4. Luodaan sd-kortti -hakemistoon uusi hakemisto, jonka nimeksi asetetaan boot.

```
mkdir boot
```

5. Kopioidaan boot-hakemistoon sulautetun Linux-käyttöjärjestelmän ydin uImage.

```
cp /home/user/Desktop/buildroot-2011.05/output/images/uImage boot
```

6. Varmistetaan että SD-kortti pitää nyt sisällään sulautetun Linux-käyttöjärjestelmän, jonka jälkeen kortti voidaan poistaa turvallisesti työasemalta.

Tehtävä 3. Kehitysalustan käynnistäminen SD-kortilta

1. Yhdistetään kehitysalusta työasemalle RS232-kaapelilla.
2. Käynnistetään työasemalla terminaaliohjelma seuraavilla asetuksilla:

Port:	/dev/ttyS0
Speed:	115200
Parity:	none
Bits:	8
Stopbits:	1
Flowcontrol:	none

3. Kytketään SD-kortti alustalle ja asetetaan alustalle virta päälle. Pysäytetään alustan käynnistyminen kohdassa:

```
Press SPACE to abort autoboot in 1 seconds
```

Jos käynnistys on kerinnyt mennä jo tämän kohdan ohitse (menee ohitse noin sata prosenttisesti), alusta käynnistetään udelleen ja käynnistys pysäytetään tässä kohdassa.

4. Välilyöntiä painamalla siirrytään Uboottiin, jossa syötetään seuraavat komennot:

```
Uboot> askenv bootcmd
Please enter 'bootcmd':mmcinit; ext2load mmc 0:1 0x10400000 /boot/uImage; bootm
Uboot> set bootargs 'console=ttyS0 root=/dev/mmcblk0p1 rootwait'
Uboot> saveenv
Uboot> boot
```

Näillä komennoilla alusta käynnistyy SD-kortilta. Käynnistämisessä menee hetki joten odotellaan maltillisesti että alustan käynnistys on valmistunut. Alustalle kirjaudutaan sisään root-käyttäjällä, salasana on oletuksena tyhjä. Alusta ohjautuu root-hakemistoon, joka on tyhjä. Root-hakemistosta voidaan siirtyä juurihakemistoon komennolla cd /.

Kehitysalusta on nyt käynnissä ja käyttää sulautettua Linux-käyttöjärjestelmää. Käyttöjärjestelmän hakemistorakenteessa voidaan navigoida normaaleilla Linux-komennoilla.

Osa 2. Tietoliikenne

Tietoliikenne tehtävissä kehitysalustan eth0-portti (WAN) kytketään Vaasan ammattikorkeakoulun verkkoon ja eth1-portti (LAN) työasemalle.

Tehtävä 1. Kehitysalustan verkkoasetukset

Kytketään kehitysalustalta virta pois päältä, poistetaan SD-kortti ja liitetään kortti työaseman kortinlukijaan. Alustan järjestelmässä on kankea käyttää tekstieditoria, joten suoritetaan tekstitiedostojen muokkaus työasemalla.

Navigoidaan kortilla hakemistoon /etc/network, josta löytyy tiedosto interfaces. Tähän tiedostoon asetetaan esitehtävien tehtävässä kaksi selvitetyt verkkoasetukset.

Tehtävä 2. DHCP-palvelin

Määritellään DHCP-palvelimen asetukset. Asetuksia varten luodaan hakemistoon /etc tiedosto dnsmasq.conf. Tässä tiedostossa määritellään DHCP-palvelimen asetukset. Asetuksissa määritellään kumpi porteista jakaa osoitteita, IP-osoitealue, maski ja aika. Määrittely onnistuu kahdella rivillä ja ohjeet löytyy osoitteesta:

http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq.con f.example.

Seuraavaksi luodaan hakemistoon /var/lib alihakemisto misc, jonka sisällä luodaan tiedosto dnsmasq.leases.

Tehtävä 3. Reititys

1. Siirretään SD-kortti kehitysalustalle ja kytketään virta päälle.
2. Kehitysalustalla oleva DHCP-palvelin asetetaan päälle komennolla:

```
dnsmsaq
```

3. Määritellään päälle IP Masquerading ja IP Forwarding. Ohjeita löytyy osoitteesta:

   http://www.knowplace.org/pages/howtos/firewalling_with_netfilter_iptables/configuration_syntax.php.

4. Tarkastele seuraavia kohtia:

Tarkastele työaseman verkkoasetuksia (ifconfig). Tarkastele työasemalta tiedostoa dhclient.leases (/var/lib/dhcp3). Tarkastele kehitysalustalta tiedostoa dnsmasq.leases. Katso mitä reittiä pitkin työaseman verkkoliikenne kulkee esim. osoitteeseen google.fi (traceroute). Katso kehitysalustalta ja työasemalta tiedosto resolv.conf.

Tehtävä 4. Siltaus

Luodaan siltaava yhteys kehitysalustan Ethernet-porttien ylitse. Siltaus luodaan bridge-utils -työkalulla. Esitehtävien tekijä on luonut siltauksen esitehtävässä numero kolme. Yksinkertaisen siltauksen luontiin tarvitaan kolme komentoa. Siltaus asetetaan päälle komennolla:

ifconfig siltauksen_nimi up

Miten työaseman verkkoasetukset ovat muuttuneet? (ifconfig)

Lopuksi poista siltaus.

Osa 3. Sulautettu web-palvelin

Tässä osiossa kehitysalustan ei tarvitse olla kytkettynä Vaasan ammattikorkeakoulun verkkoon, se riittää, että kehitysalusta ja työasema ovat kytkettynä toisiinsa. SD-kortti on jälleen hyvä liittää työasemalle, jotta tekstitiedostojen muokkaus onnistuu joustavammin. Eli kytketään alustalta virta pois päältä ja kytketään SD-kortti työasemalla.

1. Navigoidaan SD-kortilla etc-hakemistoon ja luodaan sinne alihakemisto lighttpd.
2. Luotuun hakemistoon luodaan tiedostot lighttpd.conf, access.log ja error.log.
3. Määritellään ligttpd.conf -tiedostossa seuraavat määrittelyt:

```
server.document-root = "/etc/lighttpd"
server.port = 80
server.bind = "10.0.0.1"

server.errorlog = "/etc/lighttpd/error.log"
accesslog.filename = "/etc/lighttpd/access.log"

server.modules = (
"mod_accesslog"
)

mimetype.assign = (
".html" =>"text/html"
)
index-file.names = ("html-tiedosto.html")
```

4. Luodaan web-palvelimelle html-sivusto, jonka nimeksi asetetaan haluttu nimi (html-tiedosto.html). Sivuston sisältö voi olla halutunlainen, mutta voidaan myös käyttää seuraava sisältöä:

```
<html>
<head>
<title>Tervetuloa!</title>
</head>
<body>
Tervetuloa NGW100-kehitysalustan html-sivustolle.
</body>
</html>
```

5. Web-palvelin on nyt valmiina, joten kytketään SD-kortti kehitysalustalle ja käynnistetään DHCP-palvelin ja web-palvelin. Web-palvelin käynnistetään komennolla:

```
lighttpd -f lighttpd.conf
```

6. Otetaan työasemalta selaimella yhteys lighttpd.conf -tiedostossa määriteltyyn ip-osoitteeseen. Nyt selaimelle pitäisi avautua juuri luotu html-sivusto.

Osa 4. SSH-yhteys

Tässä osiossa käytetään samanlaista kytkentää kuin osiossa kolme. Kehitysalustalle muodostetaan aluksi normaali SSH-yhteys, jonka jälkeen SSH-yhteyteen lisätään vielä todentaminen. Tässä osiossa SD-korttia ei tarvitse irrottaa kehitysalustalta, vaan alusta voidaan kokoajan pitää käynnissä.

Tehtävä 1. SSH-yhteyden muodostaminen

1. Tarkistetaan että työasemalta löytyy sovellus openssh-client
2. Tarkistetaan myös, että mikä päivämäärä kehitysalustalla on tällä hetkellä. Tarkistaminen onnistuu komennolla date. Jos päivämäärä ei ole oikea, muutetaan se oikeaksi komennolla:

```
date KKPPTTMMVVVV
```

K = Kuukausi, P = Päivä, T = Tunti, M = Minuutti ja V = Vuosi.

3. Luodaan kehitysalustalle normaali käyttäjä ja käyttäjälle salasana ja oma kotihakemisto komennolla:

```
adduser -h user user
```

4. Otetaan SSH-yhteys työasemalta kehitysalustalle normaaliin käyttäjään komennolla:

```
ssh user@10.0.0.1
```

5. Tarkastele työasemalta tiedostoa known_hosts (/home/user/.ssh)

Tehtävä 2. Todentaminen

1. Luodaan RSA-avainpari. Avainpari luodaan työasemalla. Avainparissa on julkinen ja yksityinen avain, josta yksityinen avain suojataan salasanalla. Avainpari luodaan komennolla:

```
ssh-keygen -t rsa
```

Käytetään avaimille oletusnimiä id_rsa ja id_rsa.pub ja oletushakemistoa /home/user/.ssh.

2. Siirrytään käyttämään kehitysalustaa normaalilla käyttäjällä komennolla:

```
su user
```

Siirrytään käyttäjän kotihakemistoon. Käyttäjän kotihakemistoon luodaan alihakemisto .ssh, jonne luodaan tyhjä tiedosto, jonka nimeksi asetetaan authorized_keys.

3. Kopioidaan asiakkaan julkinen avain työasemalta kehitysalustalle authorized_keys -tiedostoon komennolla:

```
scp /home/user/.ssh/id_rsa.pub user@10.0.0.1:/user/.ssh/authorized_keys
```

4. Kerrotaan SSH-palvelimena toimivalle kehitysalustalle, että se etsii julkista avainta normaalilta käyttäjältä, koska oletuksena se etsii julkista avainta root-käyttäjältä. Palvelimen asetukset määritellään etc-hakemistossa löytyvällä sshd_config -tiedostolla. Tässä tiedostossa pitää ottaa kommenteista pois yksi rivi ja asettaa siihen oikea polku seuraavalla tavalla:

```
AuthorizedKeysFile    /user/.ssh/authorized_keys
```

5. Varmistetaan että kaikki vaiheet on tehty, jonka jälkeen otetaan SSH-yhteys työasemalta kehitysalustalle.

Tehtävä 3. SSH-tunnelointi

1. Viimeisenä testataan SSH-tunnelointia kehitysalustalle. Syötetään työasemalla komento:

```
ssh user@10.0.0.1 -L localhost:8080:10.0.0.1:80
```

2. Avataan työasemalla selain ja kirjoitetaan osoiteriville:

```
localhost:8080
```

Osa 5. Työn lopetus

Kun työ on suoritettu ja raporttia varten tarvittavat tiedot ovat tallessa, voidaan työ lopetella. Käynnistetään kehitysalusta uudelleen ja siirrytään Uboottiin. Ubootissa kehitysalusta asetetaan käynnistymään sen omalla muistilla olevalla oletusjärjestelmällä. Tämä tehdään komennoilla:

```
Uboot> askenv bootcmd
Please enter 'bootcmd':fsload; bootm
Uboot> set bootargs ‘console=ttyS0 root=/dev/mtdblock1 rootfstype=jffs2’
Uboot> saveenv
Uboot> boot
```

Kytketään kehitysalustalta virta pois ja kytketään SD-kortti työasemalla kortinlukijaan. Poistetaan työn aikana luotu sd-kortti -hakemisto ja alustetaan SD-kortti fdisk-työkalulla.