

# Ubuntu-palvelimen koventaminen

Markus Ijäs

3. marraskuuta 2020

## Sisällys

<b>I</b>	<b>Perusasetukset kohdilleen</b>	<b>3</b>
<b>1</b>	<b>Käyttäjätunnukset</b>	<b>3</b>
1.1	Käytä kunnollisia salasanoja . . . . .	3
1.2	Tarkista, että käyttäjä-id 0 on vain root-käyttäjällä . . . . .	3
1.3	Tarkista, että kaikilla käyttäjillä on salasana . . . . .	4
<b>2</b>	<b>SSH</b>	<b>4</b>
2.1	Estä <i>root</i> -käyttäjän kirjautuminen . . . . .	4
2.2	Salli vain tarpeelliset käyttäjät ja ryhmät . . . . .	4
2.3	Lyhennä kirjautumisen aloitusaikaa . . . . .	5
2.4	Estä salasanalla kirjautuminen . . . . .	5
2.5	Estä ilman salasanaa kirjautuminen . . . . .	5
2.6	Rajoita kirjautumisyritysten määrää . . . . .	5
2.7	Poista X11:n välitys käytöstä . . . . .	6
2.8	Poista Rhosts-tunnistautuminen käytöstä . . . . .	6
2.9	Muita jippoja ja huomionarvoisia asioita . . . . .	6
<b>3</b>	<b>Palomuuraus (iptables)</b>	<b>7</b>
3.1	Iptablesin perusteet . . . . .	7
3.2	Taulut ja ketjut . . . . .	7
3.3	Palomuurisäännöt . . . . .	8
3.3.1	Sääntöjen listaaminen . . . . .	8
3.3.2	Yhteyksien pito avoinna . . . . .	8
3.3.3	Ping . . . . .	9
3.3.4	Tulevan liikenteen salliminen . . . . .	9
3.3.5	Muun tulevan liikenteen estäminen . . . . .	10
3.3.6	Sääntöjen tallentaminen ja palauttaminen . . . . .	10
3.3.7	Sääntöjen säilyttäminen uudelleenkäynnistyksessä . . . . .	10
3.3.8	Säännön poistaminen . . . . .	11

<b>4</b>	<b>Jaetun muistin suojaaminen</b>	<b>11</b>
<b>5</b>	<b>Turvallisuuspäivitysten automatisointi</b>	<b>12</b>
<b>II</b>	<b>Ohjelmistokohtainen koventaminen</b>	<b>12</b>
<b>6</b>	<b>Fail2ban</b>	<b>12</b>
<b>7</b>	<b>Apache</b>	<b>13</b>
7.1	Apachen ajaminen rajoitettuna käyttäjänä . . . . .	14
7.2	Tarpeettomien moduulien käytöstä poisto . . . . .	14
7.3	Lähtevien yhteyksien rajoitus . . . . .	14
7.4	apache2.conf . . . . .	14
7.5	conf-enabled/security.conf . . . . .	15
7.6	Mod_security ja mod_evasive . . . . .	15
7.7	SSL . . . . .	15
7.8	VirtualHost-esimerkki . . . . .	15
7.9	Muuta huomioitavaa . . . . .	16
<b>8</b>	<b>PHP</b>	<b>16</b>
8.1	Kovennettavat tuotantoasetukset . . . . .	16
8.2	Muut huomioitavat tuotantoasetukset . . . . .	17
<b>9</b>	<b>MariaDB</b>	<b>17</b>

## Aluksi

Ubuntun oletusasetukset ovat nykyään varsin turvalliset, mutta on silti muutama asia, jotka tulee internetissä sijaitsevalle palvelimelle tehdä ennen käyttöönottoa. Pyrin käsittelemään nämä asiat tässä oppaassa siten, että lopputuloksena on riittävän turvallinen palvelin julkiseen internetiin liitettäväksi. Opas on järjestetty pääosiltaan tehtävien tärkeysjärjestyksen ja toissijaisesti kirjoittajan mieltymysten mukaan.

Tässä oppaassa oletan, että olet ymmärtänyt asentaa Ubuntusta (tai Debianista, pätee tämä ohje siihenkin) vain minimaalisen asennuksen ainoastaan tarpeellisilla paketeilla. Oletan myös, että osaat linuxin peruskäytön komentokehotteesta (ml. `sudo` käyttö tarvittaessa), joten yksityiskohtaisten ohjeiden sijaan annan pelkkiä tarpeellisia komentoja muistin tueksi.

Oppaan esimerkit on otettu pääosin oppaan laatimisen yhteydessä pysäytetystä Ubuntu Server 20.04 LTS -asennuksesta, jossa ajetaan OpenSSH ja Pi-hole-sovelluksia. Soveltuvien osien oppaan esimerkkeinä on käytetty myös oppaan työstämiseen käytetyssä tietokoneessa ajettuja komentoja.

## Osa I

# Perusasetukset kohdilleen

Aloitetaan perusasetusten määrittelymisellä. Nämä määrittelyt on soveltuville osin syytä tehdä jokaiselle palvelimelle, ja moni näistä soveltuu myös tavalliselle koti- tai työkäytössä olevalle tietokoneelle. Perusasetuksiin sisältyy käyttöjärjestelmän asetusten ja oletuspakettien lisäksi myös SSH:n koventaminen, sillä oletan sen olevan käytössä lähes millä tahansa palvelimella.

## 1 Käyttäjätunnukset

Älä käytä root-käyttäjätunnusta palvelimelle kirjautumiseen tai oikeastaan mihinkään muuhunkaan, ellei se ole aivan välttämätöntä. Sen sijaan luo itsellesi uusi käyttäjätunnus, jota käytät palvelimen ylläpitoon. Useimmissa Linux-asennusvelhoissa tämän tekemistä suositellaankin oletuksena. Perusajatuksena on, että jokaisella palvelimen käyttäjätunnuksella on vain ne oikeudet, jotka se välttämättä tarvitsee, ja toisaalta, että jokaisella käyttäjällä on oma käyttäjätunnuksensa.

### 1.1 Käytä kunnollisia salasanoja

Kunnollinen salasana on satunnaisesti luotu vähintään 20-merkkinen merkkijono. Huomaa, ettei salasanaa tarvitse syöttää usein, joten sen ei tarvitse olla helposti muistettava. Salasanoja ei tule uudelleenkäyttää, vaan sekä pääkäyttäjälle (*root*), että muillekin tarpeellisille käyttäjille on luotava uniikit salasanat. Salasanat kannattaa tallentaa salasanamanageriin niiden käytön helpottamiseksi.

Käyttäjät voidaan pakottaa valitsemaan kunnollinen salasana PAM:in `pwquality`-komponentilla. Asennetaanpa se ensin:

```
apt install libpam-pwquality
```

`Pwquality`-komponentin asetuksia muokataan tiedostossa `/etc/security/pwquality.conf`. Vähin mitä kannattaa tehdä, on asettaa salasanan pituudelle hieman oletusta suurempi vaatimus attribuutilla `minlen = 20` (`#` merkitsee kommenttia, joten muista poistaa se rivin alusta). Otetaan tämän jälkeen asetukset käyttöön ajamalla komento `pam-auth-update` ja valitsemalla `pwquality` käyttöön. Huomaa myös, että root-käyttäjä voi vaihtaa salasanansa miksi tahansa näistä rajoitteista huolimatta, mutta tavalliselta käyttäjältä tämä ei onnistu. Se riittää.

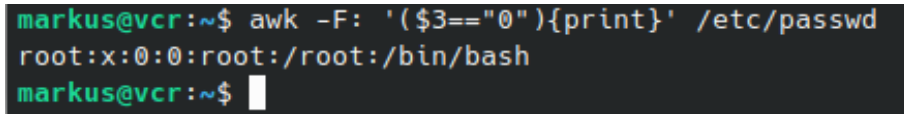
### 1.2 Tarkista, että käyttäjä-id 0 on vain root-käyttäjällä

Uudessa asennuksessa näin oletuksena onkin, mutta asia on tarvittaessa helppo selvittää oheisella pätkällä konsolissa (listaa ne käyttäjätunnukset

joiden uid on 0):

```
awk -F: '($3=="0"){print}' /etc/passwd
```

Esimerkkituloste ajettuna tämän dokumentin työstökoneessa kuvassa 1.



```
markus@vcr:~$ awk -F: '($3=="0"){print}' /etc/passwd
root:x:0:0:root:/root:/bin/bash
markus@vcr:~$
```

Kuva 1: Esimerkki käyttäjäid:n 0 tarkistuksesta awk-komennolla.

### 1.3 Tarkista, että kaikilla käyttäjillä on salasana

Cat- ja awk-komentoja sopivasti hyödyntämällä voidaan listata käytännöllisesti kaikki ne käyttäjätunnukset, joiden salasana on asetettu tyhjä merkijono (eli salasanaa ei ole):

```
cat /etc/shadow | awk -F: '($2==""){print $1}'
```

## 2 SSH

SSH on mitä todennäköisimmin käytössä palvelimellasi, joten sen koventaminen käsitellään jo tässä. Oletuksena käytetään OpenSSH-pakettia, joten siihen tämä opaskin keskittyy. OpenSSH:n asetustiedosto on `/etc/ssh/sshd_config`, johon kaikki tämän kohdan muutokset tehdään. Muutosten jälkeen on vielä muistettava käynnistää SSH-palvelu uudelleen komennolla `service ssh restart`.

### 2.1 Estä *root*-käyttäjän kirjautuminen

Estetään *root*-käyttäjän kirjautuminen, sille kun ei ole mitään tarvetta. Kaikki tarpeelliset asiat voidaan suorittaa `sudo`:lla tai korottamalla käyttäjätunnus `su`:lla. Muutetaan siis riville `PermitRootLogin` määreeksi `no`.

```
PermitRootLogin no
```

### 2.2 Salli vain tarpeelliset käyttäjät ja ryhmät

Kannattaa myös harkita vain tiettyjen käyttäjien kirjautumisen sallimista. Se tapahtuu näin:

```
AllowUsers markus@192.168.20.70 veera
```

Eri käyttäjät erotellaan siis välilyönnin, ja käyttäjän nimeen voi liittää osoitteen, josta kirjautuminen sallitaan. Esimerkissä käyttäjän *markus* kirjautuminen sallitaan ip-osoitteesta *192.168.20.70*, kun taas käyttäjän *veera* kirjautuminen mistä tahansa osoitteesta sallitaan.

Kokonaisia käyttäjäryhmiä voidaan vastaavasti sallia direktiivillä **AllowGroups**, kuten esimerkissä alla:

```
AllowGroups sshusers
```

Tällöin voit esimerkiksi luoda ryhmän **sshusers** (komennolla **addgroup sshusers**), lisätä siihen olemassa olevia käyttäjiä komennolla **usermod -a -G sshusers markus**, ja sallia koko ryhmän kerralla.

## 2.3 Lyhennä kirjautumisen aloitusaikaa

Käyttäjän palvelimelle yhdistämisen ja kirjautumisen aloittamisen välistä aikaa kannattaa myös lyhentää. Se tapahtuu pienentämällä **LoginGraceTime**-direktiivin arvoa esimerkiksi kahteenkymmeneen oletusarvosta (120).

## 2.4 Estä salasanalla kirjautuminen

Salasanalla kirjautuminen kannattaa estää, mikäli vain mahdollista. Salasanan sijaan tällöin käytetään julkisen avaimen menetelmää. Tässä kohden on huomioitava, että julkisella avaimella kirjautumisen *on onnistuttava* ennen, kuin salasanalla kirjautuminen voidaan estää. Julkisen avaimen käyttöön-otossa kannattaa turvautua **Ubuntun oppaaseen**. Kun julkisella avaimella kirjautuminen onnistuu, on käännettävä täppä **PasswordAuthentication** asentoon **no**. Tämä estää salasanalla kirjautumisen.

```
PasswordAuthentication no
PubkeyAuthentication yes
```

## 2.5 Estä ilman salasanaa kirjautuminen

Vaikka salasanalla kirjautuminen estetään edellisessä kohdassa, kannattaa erikseen estää vielä tyhjällä salasanalla kirjautuminen. Oletuksena se on estetty, mutta asia on jälleen hyvä tarkistaa (ja jos kirjautuminen on sallittu, tarkista millä käyttäjätunnuksilla ei ole salasanaa...).

```
PermitEmptyPasswords no
```

## 2.6 Rajoita kirjautumisyritysten määrää

Kirjautumisyritysten määrää on syytä rajoittaa brute-force-yritysten torjumiseksi. Tämän lisäksi kirjautumisyrityksiä kannattaa seurata jollakin sopivalla järjestelmällä (esim. Fail2Ban), ja estää epäilyttävä toiminta automaati-

tisesti jo palomuurissa. Toisena mainitsemisen arvoisena asiana on **SIEM-järjestelmän** käyttö, mutta se ei kuulu tämän oppaan laajuuteen. Alla esimerkki sopivasta rajoitteesta:

```
MaxAuthTries 3
```

Huomaa myös, ettet aseta tähän liian pientä arvoa, sillä jokainen erityyppinen kirjautumisyritys (julkinen avain, salasana jne) kuluttavat aina yhden yrityksen kukin.

## 2.7 Poista X11:n välitys käytöstä

X11-tiedonvälitysprotokollan välitys SSH:n yli kannattaa poistaa käytöstä. X11:n välitystä käytetään graafisen käyttöliittymän välitykseen SSH:n yli, ja tälle on harvemmin palvelimessa tarvetta. Tarpeeton X11:n välitys avaa yhden mahdollisen hyökkäysvektorin, joten se on hyvä pitää oletuksena pois päältä.

```
X11Forwarding no
```

## 2.8 Poista Rhosts-tunnistautuminen käytöstä

Rhosts on käytännössä eräs SSH:ta edeltävistä tavoista tunnistautua ja suorittaa komentoja etäpalvelimella. Se sisältää useita tietoturva- ja käytettävyysongelmia [12], joita ei tarkemmin tässä oppaassa käydä, joten sen käyttö tunnistautumiseen SSH:n yli kannattaa poistaa käytöstä:

```
IgnoreRhosts yes
```

Tämän pitäisi nykyään olla oletuksena näin, mutta kannattaa silti tarkistaa asia.

## 2.9 Muita jippoja ja huomionarvoisia asioita

SSH:n käyttämän **portin** vaihtamista kannattaa harkita, vaikkei se itsessään lisää tietoturvaa. Hyöty epätavallisen portin (esim. 486) valinnasta on lähinnä hieman pienempi porttiin kohdistuvien automaattisten hyökkäyssovellusten tekemä kirjautumisyritysten määrä (ja siten hieman pienempi lokikirjausten määrä). Huomaa, ettet aseta SSH:n porttia päällekkäin minäkään toisen palvelimella ajettavan palvelun kanssa, ja että muistat asettaa palomuurisäännöt uuden portin mukaiseksi. Esimerkiksi:

```
Port 486
```

SSH:n käyttämä **protokolla** kannattaa myös tarkistaa. Sen pitäisi olla (tämän oppaan kirjoitushetkellä) versiota 2. Konfiguraatietiedostossa käytettävä protokolla määritellään yksinkertaisesti: `Protocol 2`.

SSH:lla kirjautuvien käyttäjien käytettävissä olevia **komentoja** voidaan rajoittaa käyttäjäkohtaisesti tämän `authorized_keys`-tiedostossa käytettävällä `command`-optiolla. Yksinkertaisimmillaan yhden sallitun komennon määrittely avainkohtaisesti tapahtuu kuten tässä esimerkissä (sallitaan `ps`-komento, ei muuta, kirjoita yhdelle riville):

```
command="ps " ,no-agent-forwarding ,no-port-forwarding ,  
no-x11-forwarding AVAIMEN_TYYPPI AVAIN KOMMENTTI
```

## 3 Palomuuraus (iptables)

### 3.1 Iptablesin perusteet

Iptablesia käytetään Linuxin ytimen IP-pakettisuodattimen (toteutettu *Netfilter*-moduulein) sääntöjen asetukseen, muokkaukseen ja ylläpitoon. Netfilterin säännöt on järjestetty *tauluissa* (*table*) sijaitseviin *ketjuihin* (*chain*), joihin kootaan olennaiset suodatussäännöt. Käsittelen iptablesin vain lyhyehkösti, joten kannattaa tutustua iptablesin man-sivun lisäksi myös lähdeluettelossa linkitettyyn materiaaliin. Iptables, kuten moni muukin tässä oppaassa, tarvitsee korotetut oikeudet toimiakseen, joten sitä on hyvä komentaa `sudo` hyväksi käyttäen.

Iptables on tarkoitettu IPv4-protokollaa koskeville säännöille. IPv6-protokollalle voidaan puolestaan käyttää `ip6tables`ia. Muita eri käyttötarkoituksiin luotuja ovat `arptables` ja `ebtables`. Iptablesia voidaan ohjata myös UFW-paketilla, mutta tämä ei todellisuudessa juuri helpota palomuurisääntöjen luomista, joten sitä ei käsitellä tässä.

### 3.2 Taulut ja ketjut

Netfilterillä on tavanomaisesti neljä taulua: *nat*, *filter*, *mangle* ja *raw*. Näitä tauluja käytetään seuraavasti:

**Nat** -taulua käsittelemään sellaisia paikallisen koneen läpi haluava paketteja, joiden tarkoitus on jatkaa matkaansa järjestelmästä ulos.

**Filter** -taulua pääättelemään, saako paketti jatkaa matkaansa kohteeseensa.

**Mangle** -taulun sääntöjen avulla voidaan muokata paketin otsikkotietoja *header*.

**Raw** -taulua pääasiallisesti poikkeuksien tekemiseen ennen muuta reititystä.

Meitä kiinnostaa tämän oppaan puitteissa eniten filter-taulu, joten keskitymme siihen. Filter-tauluun valmiiksi luodut ketjut ovat *input*, *forward* ja *output*. Niiden tehtävät ovat:

**Input** käsittelee paikalliseen koneeseen *tulevia* paketteja.

**Forward** käsittelee paikalliseen koneeseen tulevia eteenpäin *välitettäviä* paketteja.

**Output** käsittelee paikallisesta koneesta *lähteviä* paketteja.

Muissa tauluissa on vastaavasti niiden taulujen tarvitsemat ketjut palomuurisäännöille.

### 3.3 Palomuurisäännöt

Yksinkertaisimmillaan palomuurin tulisi sallia vain välttämätön sisäänpäin tuleva liikenne, ja estää kaikki muu liikenne sisäänpäin. Ulos lähtevän liikenteen voi oletuksena sallia, mutta poikkeuksia voi siihen käytäntöön olla.

#### 3.3.1 Sääntöjen listaaminen

Filter-listalla olevat säännöt näkee komennolla

```
iptables -t filter -L
```

jossa määre **-t filter** käskee tehdä haun filter-listalle, ja **-L** käskee näyttää kaikki säännöt. Voit halutessasi lisätä komenttoon vielä määreen **-v**, joka käskee listaamaan säännöt tavallista tarkemmin. Mikäli määrettä **-t** ei anneta, koskevat iptablesille annetut komennot oletuksena filter-listaa. Tämän vuoksi määrettä ei jatkossa tässä oppaassa anneta, ellei se ole erityisesti tarpeellista. Vastaavasti IPv6-säännöt näkee komennolla **ip6tables -t filter -L**.

#### 3.3.2 Yhteyksien pito avoimna

Ensimmäiseksi tulee pitää jo avatut yhteydet avoimina, jottet epähuomiossa lukitse itseäsi ulos palvelimelta heti ensitöiksesi, ja toisaalta jotta palvelimelta ulospäin lähtevät yhteydet toimivat oikein. Lisäksi tulee sallia yhteydet lo-rajapintaan.

Nykyään osan **-m conntrack -ctstate** sijaan riittäisi myös **-m state -state**, sillä moderneissa Linuxin ytimissä tämä **-m state** todellisuudessa ohjataan **conntrack**-moduulille. Esimerkissä käytetään kuitenkin edelleen aiempaa muotoa taaksepäin yhteensopivuuden varmistamiseksi.

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m conntrack --ctstate
    ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A INPUT -m conntrack --ctstate
    ESTABLISHED,RELATED -j ACCEPT
```



Tämä on samalla iptablesin palomuurisääntöjen asettamisen perussyntaksi, joten käydäänpä läpi annetut määreet:

**-A INPUT** Lisätään sääntö input-ketjun viimeiseksi.

**-m conntrack** Ottaa käyttöön conntrack-moduulin sallien yhteyden tilan tarkemman seurannan.

**-ctstate ESTABLISHED,RELATED** Määrittää säännön koskemaan vain sellaisia paketteja, jotka liittyvät johonkin aiempaan yhteyteen. (huomaa määreen edessä kaksi viivaa, jotka  $\text{\LaTeX}$ vääntää tässä väärään muotoon)

**-j ACCEPT** Sallitaan tähän sääntöön täsmäävien pakettien tulo palvelimelle.

### 3.3.3 Ping

Saatat haluta sallia pingiin vastaamisen. Pingiin vastaamiseksi tarvitaan seuraavat säännöt:

```
iptables -A INPUT -p icmp --icmp-type echo-request
-j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request
-j ACCEPT
```

### 3.3.4 Tulevan liikenteen salliminen

Tarpeellinen palvelimelle tuleva liikenne on tietenkin sallittava. Salliminen kannattaa periaatteessa toteuttaa mahdollisimman tiukoin säännöin, jotta välttyään sallimasta asiatonta liikennettä vahingossa. Sallitaan esimerkiksi kaikki sisään tuleva tcp-protokollan mukainen liikenne porttiin 486, joka valittiin aiemmin SSH-palvelimen portiksemme:

```
iptables -A INPUT -p tcp --dport 486 -j ACCEPT
ip6tables -A INPUT -p tcp --dport 486 -j ACCEPT
```

Uusina määreinä tässä komennossa ovat:

**-p tcp** Määritellään sääntö koskemaan tcp-protokollan mukaista liikennettä.

**--dport** Määritellään lisäksi sääntö koskemaan vain porttiin 486 suuntautuvaa liikennettä.

Esimerkkipalvelimemme tapauksessa haluamme lisäksi sallia tulevan udp-liikenteen porttiin 53 ja tulevan tcp-liikenteen porttiin 80:

```
iptables -A INPUT -p udp --dport 53 -j ACCEPT
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
ip6tables -A INPUT -p udp --dport 53 -j ACCEPT
ip6tables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Huomaa, että yllä oleva on vain esimerkki, eikä mahdollista vielä Pi-holen täydellistä toimintaa. Pi-holen palomuuriasetuksia koskien kannattaa tutustua heidän [asennusvaatimuksiinsa](#).

### 3.3.5 Muun tulevan liikenteen estäminen

Palomuurista on kovin vähän hyötyä, mikäli se sallii kaiken liikenteen sisään-päin. Tämän vuoksi sääntöketjun loppuun onkin lisättävä kaiken liikenteen estävä sääntö:

```
iptables -A INPUT -j DROP
ip6tables -A INPUT -j DROP
```

Tämä sääntö nappaa kiinni sellaisen liikenteen, joita aikaisemmat säännöt eivät koske (eli jota ei haluta palvelimelle tulevan)

### 3.3.6 Sääntöjen tallentaminen ja palauttaminen

Sääntölistausta voidaan tallentaa säilyttäen tavu- ja pakettilaskurit (-c):

```
iptables-save -c > {{polku/ja/nimi}}
```

Ja palauttaa vastaavasti:

```
iptables-restore < {{polku/ja/nimi}}
```

### 3.3.7 Sääntöjen säilyttäminen uudelleenkäynnistyksessä

Iptablesin säännöt eivät säily oletuksena palvelimen uudelleenkäynnistyksen yli. Helpoiten säännöt saa säilymään asentamalla **iptables-persistent**-paketin. Asennuksen yhteydessä paketti konfiguroidaan. Oletusasetukset ovat hyvät (eli vastaa kyllä molempiin kysymyksiin).

Säännöt tallentuvat tiedostoon `/etc/iptables/rules.v4`. Kun teet muutoksia sääntöihin ensiasennuksen jälkeen, muista päivittää tämä tiedostot lopuksi komennoilla

```
iptables-save -c -f /etc/iptables/rules.v4
ip6tables-save -c -f /etc/iptables/rules.v6
```

Sääntöjen säilyminen kannattaa koestaa käynnistämällä palvelin uudelleen, ja tarkistamalla käytössä olevat säännöt sitten komennolla **iptables -L**.

### 3.3.8 Säännön poistaminen

Jos teit sääntöjä tehdessäsi virheen, voit poistaa sen helpoiten listaamalla ensin säännöt numeroituina lisäämällä peruslistauskomentoon määreen `--line-numbers`:

```
iptables -L --line-numbers
TAI ip6tables -L --line-numbers
```

Säännön poistaminen numeroa käyttäen tapahtuu `-D`-määreen avustuksella seuraavasti:

```
iptables -D {{ketju}} {{rivin_numero}}
TAI ip6tables -D {{ketju}} {{rivin_numero}}
```

Esimerkkituloste ajettuna tämän dokumentin työstökoneessa alla (tulostetta on hieman muokattu paremmin tälle sivulle sopivaksi):

```
# iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1          all  —  anywhere anywhere    state REL...
2  ACCEPT  all  —  anywhere anywhere    state REL...
```

```
# iptables -D INPUT 1
```

```
# iptables -t filter -L --line-numbers
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1  ACCEPT  all  —  anywhere anywhere    state REL...
```

Input-ketjun rivi 1 on siis poistettu, jolloin aiemmasta rivistä 2 tulee tietenkin uusi rivi 1.

## 4 Jaetun muistin suojaaminen

Oletuksena `/run/shm` (käytännössä muistin jaettu osuus) asetetaan käyttöön luku- ja kirjoitusoikeuksin, sekä oikeuksin suorittaa ohjelmia. Tavalliselle työasemalle tämä on joskus jopa toivottua, mutta palvelimella ei niinkään. Jaettua muistia voidaan käyttää hyökkäyksiin käynnissä olevia ohjelmistoja vastaan, joten se on suojattava. Suojaaminen tapahtuu lisäämällä `/etc/fstab`-tiedoston loppuun rivi:

```
none /run/shm tmpfs defaults,ro 0 0
```

Tämä liittää `/run/shm`:n käyttöön vain luettavassa muodossa tietokoneen käynnistyksen yhteydessä. Mikäli tämä aiheuttaa ongelmia (ei pitäisi yksinkertaisissa palvelimissa), voit myös tehdä liittämisen *yllä olevan sijaan* seuraavasti:

```
none /run/shm tmpfs rw,noexec,nosuid,nodev 0 0
```

Muutokset otetaan käyttöön seuraavan uudelleenkäynnistyksen yhteydessä, mutta halutessasi voit ottaa ne käyttöön heti komennolla `mount -o remount /run/shm`.

## 5 Turvallisuuspäivitysten automatisointi

Turvallisuuspäivitysten automaattiseen asentamiseen kannattaa käyttää pakettia `unattended-upgrades`, mikäli automatisointi ei ole valittavissa jo asennusvaiheessa. Asennus tapahtuu tavalliseen tapaan komennolla:

```
apt install unattended-upgrades
```

Unattended-upgrades-paketin konfigurointia varten kannattaa tutustua [Linux Auditin mainioon blogikirjoitukseen aiheesta](#). Lyhyesti sanoen konfigurointi tapahtuu ajamalla ensin konfigurointiskripti, ja sitten muokkaamalla tiedostoa `/etc/apt/apt.conf.d/50unattended-upgrades`. Alla esitetään tärkeimmät rivit tiivistetysti.

Ajetaan ensiksi konfigurointiskripti (vastaa kysymykseen kyllä...):

```
dpkg-reconfigure --priority=low unattended-upgrades
```

Avataan sitten yllä mainittu konfiguraatio. Aktivoidaan automaattinen uudelleenkäynnistys yöllä, ja asetetaan lokitietojen kirjoitus Syslogiin (oleellista, jos otetaan käyttöön syslogin tietojen siirto etäpalvelimelle):

```
Unattended-Upgrade::Automatic-Reboot "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";
Unattended-Upgrade::SyslogEnable "true";
```

Säännölliset uudelleenkäynnistykset ovat tärkeitä muun muassa Linuxin ytimen päivitysten loppuun saattamiseksi. Ilman uudelleenkäynnistystä ytimeen tehtyt tietoturvapäivitykset jäävät käyttöön ottamatta.

Tämän oppaan laajuuden ulkopuolelta kannattaa harkita mm. sähköpostien automaattilähetystä, jotta ongelmat päivitysten kanssa tulevat tietoon mahdollisimman nopeasti. Sen lisäksi, että automaattiset päivitykset ovat käytössä, kannattaa tarkistaa ja asentaa muutkin päivitykset säännöllisesti.

## Osa II

# Ohjelmistokohtainen koventaminen

## 6 Fail2ban

Suosittelena asentamaan etenkin julkisessa internetissä oleville palvelimille peruskoventamisen lisäksi Fail2ban-sovelluksen. Fail2ban auttaa suoja-

maan palvelinta automatisoiduilta hyökkäyksiltä tarkkailemalla lokitiedostoja ja lisäämällä automaattisesti palomuurisääntöjä estääkseen osaltaan palvelimen tarjoamiin palveluihin kohdistuvia kirjautumis- ja palvelunestoyrityksiä. Fail2ban asennetaan tuttuun tapaan komennolla **apt install fail2ban**.

Fail2ban tekee estot suoraan netfilter-palomuriin, ja poistaa estot automaattisesti asetustensa mukaisesti. Fail2banissa käytetään jail-konseptia (ns. vankiloita), jossa jokainen jail kuvastaa yhtä palvelimen palvelua ja sen suojaamistoimenpiteitä.

Fail2banin asennuksen yhteydessä asennetaan myös komentorivikäyttöliittymä fail2ban-client. Tälle on muutama tärkeä komento:

**fail2ban-client start** käynnistää Fail2ban:n.

**fail2ban-client reload** lataa konfiguraatioihin tehdyt muutokset ja uudelleenkäynnistää Fail2banin.

**fail2ban-client status** kertoo palvelun yleisilasta.

**fail2ban-client status {{jail}}** kertoo yksittäisen jailin tilanteesta.

Fail2banin asetukset löytyvät hakemistosta `\etc\fail2ban\`. Asetukset jaetaan **fail2ban.conf** ja **jails.conf** -tiedostoihin, joiden sisältämiä asetuksia korvataan `.local`-loppuisilla tiedostoilla. Itse `.conf`-tiedostoja ei tule muuttaa, sillä muutokset eivät välttämättä säily ohjelmistopäivitysten yli. Local-tiedostoihin tehdyt muutokset puolestaan säilyvät.

Yleisesti ottaen asetukset tehdään `jails.local`-tiedostoon, joten luodaan se kopioimalla `conf`-tiedoston sisältö komennolla

```
cp /etc/fail2ban/jail.{conf,local}
```

Asetusten tekeminen sinänsä jätetään pois tästä oppaasta tarkoituksellisesti, sillä se riippuu varsin paljon kunkin hetkisestä tarpeesta.

Erityishuomio kohdennettakoon kuitenkin tiedoston riviin

```
#ignoreip = 127.0.0.1/8 ::1
```

jonka kommenttimerkki kannattaa poistaa, ja johon kannattaa lisätä oma ip-osoitteesi (tai dns-osoitteesi), jotta se ei joudu vahingossa estetyksi. Toisaalta tulee huomioida jailsien muokkauksessa mahdollisesti vaihdetut portit esimerkiksi `ssh:n` osalta (fail2ban olettaa portin olevan 22).

## 7 Apache

Apache asennettuna Ubuntuun tai Debianiin on oletuksena varsin turvallinen. On kuitenkin joitain muutoksia, jotka jokaiselle palvelimelle olisi hyvä tehdä.

## 7.1 Apachen ajaminen rajoitettuna käyttäjänä

Ubuntuun asennettaessa Apachea ajetaan oletuksena käyttäjänä ja ryhmässä `www-data`. Tämä on tavallisesti riittävän turvallista, mutta tarvittaessa kannattaa harkita käyttäjänimen ja ryhmän vaihtamista toiseen.

## 7.2 Tarpeettomien moduulien käytöstä poisto

Luonnollista on, että tarpeettomat lisäosat kannattaa poistaa käytöstä ohjelmasta kuin ohjelmasta. Sama pätee Apacheen. Harkitse siis esimerkiksi seuraavien moduulien käytöstä poistamista:

- `userdir`
- `suexec`
- `cgi/cgid`
- `include`
- `autoindex`

Käytössä olevia moduulien asetustiedostoja voit tarkastella kansiossa `/etc/apache2/mods-enabled`

## 7.3 Lähteiden yhteyksien rajoitus

Lähteiden yhteyksien rajoittamista palomuurissa kannattaa harkita. Opastusta tähän saa esimerkiksi [Debianin wikin Apache Hardening -sivulta](#).

## 7.4 `apache2.conf`

Apachen perusasetukset löytyvät asetustiedostosta `/etc/apache2/apache2.conf`. Niiden muokkaamista seuraavasti kannattaa harkita.

Oletuksena Apache sallii pyynnön suorittamisen 300 sekunnin aikana, mikä saattaa altistaa palvelimen esimerkiksi Slow Loris -hyökkäykselle. Kannattaakin harkita asetustiedoston `Timeout`-määreen laskemista esimerkiksi kuuteenkymmeneen.

ETag-otsake kannattaa poistaa käytöstä, sillä se saattaa paljastaa liikaa tietoja palvelimella sijaitsevista tiedostoista. Otsake poistetaan käytöstä lisäämällä asetustiedostoon rivi `FileETag None`.

## 7.5 conf-enabled/security.conf

Apachen asetustiedostoa `/etc/apache2/conf-enabled/security.conf` muuttamalla vaikutetaan useaan tärkeään asetukseen. Ne käsitellään tässä osiossa.

Järjestelmän versiotietojen piilotus on tärkeää, sillä sen perusteella voidaan päätellä, kuinka palvelimelle kannattaa hyökätä. Piilotus tapahtuu asettamalla asetukset seuraavasti:

```
ServerTokens Prod
ServerSignature Off
```

X-Frame-Options-määreen asettaminen kannattaa tehdä Cryptojacking-hyökkäysten estämiseksi. Kannattaa myös aktivoida keksien asetus `HttpOnly`- ja `Secure`-lipuin, ja suojautua XSS-hyökkäyksiltä. Huomioi, että nämä tarvitsevat moduulin headers asettamisen käyttöön komennolla `a2enmod headers` komentokehotteessa.

```
Header set X-Frame-Options: "sameorigin"
Header edit Set-Cookie ^(.*)$ $1; HttpOnly; Secure
Header set X-XSS-Protection "1; mode=block"
```

## 7.6 Mod\_security ja mod\_evasive

Näiden käyttöä suositellaan, mutta eivät kuulu toistaiseksi oppaan laajuuteen.

## 7.7 SSL

Kaiken http-liikenteen olisi hyvä kulkea salattuna, jotta verkkosivuille syötetyt ja sieltä haetut tiedot pysyisivät siirron ajan turvassa. Erityisen tärkeää tämä on esimerkiksi kirjautumistietojen kohdalla, joiden et halua kulkevan selväkielisinä http-protokollan yli.

Sen sijaan, että kirjoittaisin asiat tähän uudelleen, ohjaan Let's Encryptin SSL-sertifikaattien käyttöönottoon esimerkiksi Linuxizen ohjeen perusteella: <https://linuxize.com/post/secure-apache-with-let-s-encrypt-on-ubuntu-20-04/>.

## 7.8 VirtualHost-esimerkki

Esimerkki suhteellisen turvallisesta VirtualHost-määrittelystä esitetään alla. Määrittelyssä kannattaa huomioida tiedostolistausten esto (`-Indexes`), protokollamäärittely ja SSL:n käyttö.

```
<VirtualHost *:443>
    ServerName markus.ihakset.name
```

```

Protocols h2 http/1.1

ServerAdmin webmaster@localhost
DocumentRoot /var/www/markus

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /var/www/markus>
    Options -Indexes
</Directory>

SSLEngine On
SSLCertificateFile /dir/to/fullchain.pem
SSLCertificateKeyFile /dir/to/privkey.pem
</VirtualHost>

```

## 7.9 Muuta huomioitavaa

<https://observatory.mozilla.org>

## 8 PHP

PHP:n ollessa käytössä on myös sen koventaminen olennaista. Tässä oppaassa kerron muutaman perusasetuksen, mutta koska PHP on varsin iso ja monimutkainen, ja tarvittavat kovennukset riippuvat tarjottavista palveluista, en voi millään käsitellä aihetta kovinkaan kattavasti.

Yleisellä tasolla PHP:n moduuleista kannattaa poistaa käytöstä kaikki tarpeettomat. Kattavaa listaa niistä ei voi tässä antaa, sillä se riippuu palvelimella tarjottavista PHP:ta käyttävistä palveluista.

PHP:n perusasetukset tehdään tiedostoon `/etc/php/7.4/apache2/php.ini`. Palvelinkonfiguraatiosta riippuen asetustiedosto voi sijaita muuallakin, mutta käyttämäni esimerkkipalvelimen tapauksessa sijainti on yllä mainittu. Nykyisellään tämä tiedosto on ns. tuotantoasetusten tiedosto, on oletuksena käytössä, ja on kehitysasetuksia turvallisempi.

### 8.1 Kovennettavat tuotantoasetukset

Listaan alla kovennettavat kohdat tuotantoasetustiedostosta, joskin oppaan muusta tyylistä poiketen vähemmällä selityksellä. Tarkemmat selitykset voit lukea suoraan PHP:n asetustiedostosta, joten niiden suomentaminen tähän on tarpeetonta. Kaikkien listattujen muutosten tekemistä on syytä harkita tarpeen mukaan.



```
file_uploads = Off
allow_url_fopen = Off
open_basedir = "/var/www:/var/lib/php/sessions"
session.use_strict_mode = 1
```

Lisäksi kannattaa poistaa muutama funktio käytöstä, jollei niille ole tarvetta lisäämällä asetuksen `disable_functions` olemassa olevien nimien perään:

```
exec , passthru , shell_exec , system , proc_open , popen ,
curl_exec , curl_multi_exec , parse_ini_file , show_source ,
```

Sessiokeksien asetteluun ei mennä yksityiskohtaisesti, vaan se on tehtävä kulloisenkin tarpeen mukaan.

## 8.2 Muut huomioitavat tuotantoasetukset

Alla listatut kohdat ovat koventamisen kannalta olennaisia edellisen kohdan listan lisäksi, mutta ovat oletuksena asetettu järkevästi. Näitä listattuja kohtia ei siis tarvitse muuttaa, mutta ne kannattaa kuitenkin tarkistaa.

```
expose_php = Off
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT
display_errors = Off
log_errors = On
allow_url_include = Off
cgi.force_redirect = 1
session.save_path = "/var/lib/php/sessions"
```

## 9 MariaDB

MariaDB:n osalta ohjaan suoraan heidän omaan oppaaseensa:

[https://mariadb.com/kb/en/mysql\\_secure\\_installation/](https://mariadb.com/kb/en/mysql_secure_installation/).

## Viitteet

- [1] Fail2ban wiki. Luettu 3.10.2020. Saatavilla: <https://www.fail2ban.org>
- [2] Gite, V. 2019. Linux 25 PHP Security Best Practices For Sys Admins. Cyberciti.biz. Luettu 5.10.2020. Saatavilla: <https://www.cyberciti.biz/tips/php-security-best-practices-tutorial.html>
- [3] How to: Make iptables rules persistent between reboots on Debian/Ubuntu 18, 19. Dannyda.com. Luettu 29.9.2020. Saatavilla: <https://dannyda.com/2020/04/29/how-to-make-iptables-rules-persistent-between-reboots-on-debian-ubuntu-18-19/>
- [4] Hainic, C. 2017. Using Iptables Instead Of Ufw In A Basic Server Setup & For Docker. Spyhce.com. Luettu 29.9.2020. Saatavilla: <https://spyhce.com/blog/using-iptables-instead-ufw-basic-server-setup>
- [5] Iptablesin man-sivu. Die.net. Luettu 29.9.2020. Saatavilla: <https://linux.die.net/man/8/iptables>
- [6] Iptablesin Wikipedia-sivu. Luettu 29.9.2020. Saatavilla: <https://en.wikipedia.org/wiki/Iptables>
- [7] Linux Audit: Ubuntu system hardening guide for desktops and servers. Luettu 26.9.2020. Saatavilla: <https://linux-audit.com/ubuntu-server-hardening-guide-quick-and-secure/>
- [8] Linuxize: How to Install and Configure Fail2ban on Ubuntu 20.04. Luettu 3.10.2020. Saatavilla: <https://linuxize.com/post/install-configure-fail2ban-on-ubuntu-20-04/>
- [9] NuHarbor Security: Ubuntu Server Hardening Guide. Luettu 26.9.2020. Saatavilla: <https://www.nuharborsecurity.com/ubuntu-server-hardening-guide-2/>
- [10] Rossi, L. 2018. How to Harden Your PHP for Better Security. Howtogetonline.com. Luettu 5.10.2020. Saatavilla: <https://howtogetonline.com/how-to-harden-your-php-for-better-security.php>
- [11] Schroder, C. 2017. How to Write iptables Rules for IPv6. Linux.com. Luettu 4.10.2020. Saatavilla: <https://www.linux.com/topic/networking/iptables-rules-ipv6/>
- [12] SSH Communications Security, Inc. rsh. Luettu 27.9.2020. Saatavilla: <https://www.ssh.com/ssh/rsh>

- [13] Ubuntu documentation: StricterDefaults. Luettu 26.9.2020. Saatavilla:  
<https://help.ubuntu.com/community/StricterDefaults>