



Ubuntu Apache2 veebiserveri kahepoolse SSL-i häälestus Eesti eID kaartide vaates

Dokumendi info	
Loomise aeg	06.02.2019
Tellijä	RIA
Autor	Urmas Vanem, OctoX
Versioon	23.02/1

Versiooni info		
Kuupäev	Versioon	Muutused/märkused
06.02.2019	19.02/1	Avalik versioon.
20.02.2019	19.02/1	Lisatud võimalike lisakonfiguratsioonide peatükk: tulemüüri ja OCSP seadistus ning vaikimisi veebisaidi eemaldamine. Muutja: Urmas Vanem
12.12.2019	19.12/1	Lisatud Apache soovituslikud turvasätted. Muutja: Urmas Vanem
16.12.2020	20.12/1	Lisatud nõue kasutajasertifikaadile omada korrektset extendedKeyUsage välja ja „õiget“ sertifikaadi väljastajat. Vt. peatükk „kasutajasertifikaatide lisafiltreerimine“. Muutja: Urmas Vanem
17.12.2020	20.12/2	Lisatud direktiiv SSLCADNRequestPath, vt. peatükki „CA-de sertifikaatide filtreerimine kliendile“. Muutja: Urmas Vanem
13.01.2021	21.01/1	Lisatud demo-konfiguratsiooni fail lingina. Lisatud HSTS konfiguratsioon. Muutja: Urmas Vanem
21.01.2021	21.01/2	Parandatud SSLOCSPEnable parameeter: on->leaf. Uuendatud TLS 1.2 <i>cipher</i> 'ite soovitusel. Uuendatud TLS protokollide kasutamise soovitusel. Demokonfi ja dokumendi muutujate nimed on sünkroniseeritud. Muutja: Urmas Vanem
27.01.2021	21.01/3	Lisatud „mobiil-id“ filter. Muutja: Urmas Vanem

Ubuntu/Apache2 SSL häälestus



Lihtne konfiguratsioonijuhend Eesti EID kaartide vaates

26.02.2021	21.02/1	Lisatud alternatiivne kesktaseme sertimiskeskuste filtreerimisvõimalus SSLCADNRequestFile direktiivi abil. Muutja: Urmas Vanem
27.04.2021	21.04/1	Eemaldatud aegunud ESTEID-SK 2011 sertifikaatide tugi. Muutja: Urmas Vanem
25.11.2021	21.11/1	Ubuntu on uuendatud versioonile Ubuntu Server 21.10 ja Apache versioonile 2.4.48. Lisatud on ECC sertifikaatide loomine veebiserveril. Täiendatud on TLS ja Cipher soovitusi. Muutja: Urmas Vanem
21.02.2023	23.02/1	Ubuntu on uuendatud versioonile Ubuntu Server 22.04 ja Apache versioonile 2.4.55. Uuendatud on ka virtuaalhosti konfiguratsiooni. Muutja: Urmas Vanem



Sissejuhatavalt

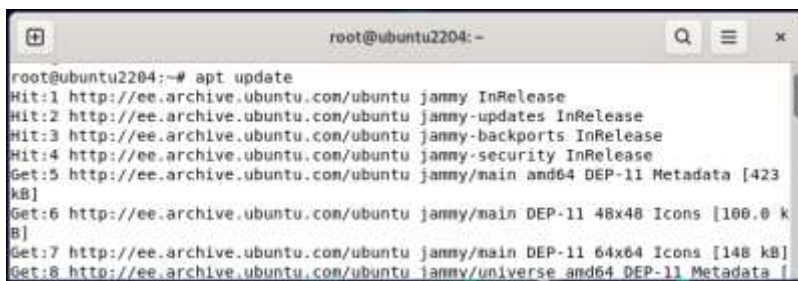
Käesolevas juhendis kirjeldame:

- Kuidas installeerida ja häälestada Apache2 (v. 2.4.55) veebiserver Ubuntu 22.04 serveril!?
- Kuidas häälestada HTTPS (ühepoolne SSL) veebiserveril!?
- Kuidas häälestada eID kaartidega autentimine (kahepoolne SSL) vastu veebiserverit!?
- Lisaks vaatame muid võimalusi serveri konfigureerimiseks ja pakume välja ka soovitusi turvalisuse vaates.

Apache2 installatsioon ja häälestus

Installatsioon

1. Uuendame Ubuntu pakke andmed käivitades terminalis käsu „apt update“.



```
root@ubuntu2204:~# apt update
Hit:1 http://ee.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ee.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ee.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://ee.archive.ubuntu.com/ubuntu jammy-security InRelease
Get:5 http://ee.archive.ubuntu.com/ubuntu jammy/main amd64 DEP-11 Metadata [423 kB]
Get:6 http://ee.archive.ubuntu.com/ubuntu jammy/main DEP-11 48x48 Icons [100.0 kB]
Get:7 http://ee.archive.ubuntu.com/ubuntu jammy/main DEP-11 64x64 Icons [148 kB]
Get:8 http://ee.archive.ubuntu.com/ubuntu jammy/universe amd64 DEP-11 Metadata [
```

Pilt 1 – pakke uuendamine

2. Installeerime Apache2-e käsuga „apt install apache2“.



```
root@ubuntu2204:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-data apache2-utils mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-data apache2-utils mailcap mime-support ssl-cert
0 upgraded, 6 newly installed, 0 to remove and 64 not upgraded.
```

Pilt 2 - Apache2 installatsioon

Eelneva tegevuse tulemusena on Apache server installeeritud¹:

¹ Tänapäev (21.02.2023) viimane Apache versioon on 2.4.55, versioon 2.4.52 on Ubuntu vaikesuuna kaasas.



```
root@ubuntu2204:~# apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-01-23T18:34:42
root@ubuntu2204:~#
```

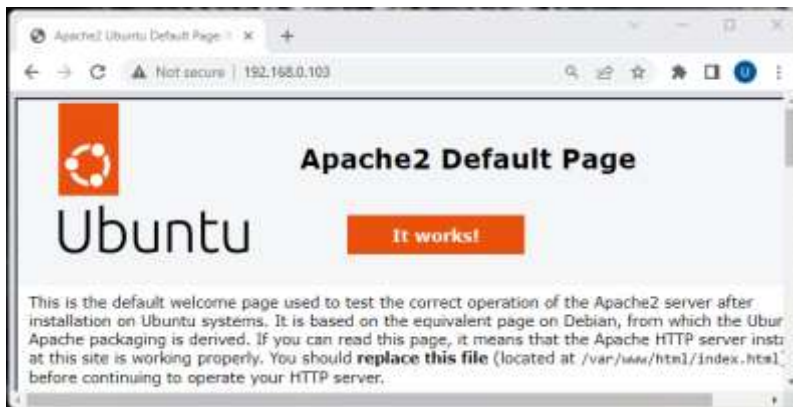
Pilt 3 - Apache versiooni päring

Uuendame Apache versioonile 2.4.55. Selleks on internetis palju juhiseid olemas² ja pikemalt me sellel protseduuril siin ei peatu.

```
root@ubuntu2204:~# apache2 -v
Server version: Apache/2.4.55 (Ubuntu)
Server built:   2023-01-19T19:55:31
root@ubuntu2204:~#
```

Pilt 4 – Apache versiooniks peale uuendamist on 2.4.55

Apache2 veebiserver versiooniga 2.4.55 töötab nüüd eaturvalises http režiimis:



Pilt 5 – Apache veebiserver vaikimisi konfiguratsioonis

Konfiguratsioon

Ühepoolse SSL-i lubamine

Lubame Apache serveril SSL mooduli käsuga „a2enmod ssl“ ja restardime Apache2 teenuse.

² Näiteks <https://ubiq.co/tech-blog/upgrade-apache-version-ubuntu/>.



```
root@ubuntu2204:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu2204:~# systemctl restart apache2
root@ubuntu2204:~#
```

Pilt 6 - SSL lubamine ja teenuse restart

Sertifikaadi privaatvõtme ja päringufaili loomine

ECC

Loome esmalt ECC algoritmil baseeruva privaatvõtme:

1. „openssl ecparam -name secp384r1 -genkey -noout -out Apache2204.key“

Ja seejärel loome privaatvõtme baasil sertifikaadi päringu CA serverile:

2. „openssl req -new -key Apache2204.key -out Apache2204.csr -subj /C=EE/O=OctoX/CN=Apache2204.octox.demo -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(printf \"[SAN]\\nsubjectAltName=DNS:Apache2204.octox.demo,DNS:MYWEBSERVER.octox.demo)\")“
3

```
uv@ubuntu2204:~/temp$ openssl ecparam -name secp384r1 -genkey -noout -out Apache
2204.key
uv@ubuntu2204:~/temp$ openssl req -new -key Apache2204.key -out Apache2204.csr -
subj /C=EE/O=OctoX/CN=Apache2204.octox.demo -reqexts SAN -config <(cat /etc/ssl/
openssl.cnf <(printf \"[SAN]\\nsubjectAltName=DNS:Apache2204.octox.demo,DNS:MYWEBS
ERVER.octox.demo\")
uv@ubuntu2204:~/temp$
```

Pilt 7 – ECC privaatvõtme ja sertifikaadi päringufaili loomine

Kollase taustaga muutujatest:

1. Apache2204.key on sertifikaadi privaatvõti;
2. Apache2204.csr on sertifikaadi päring, mis edastatakse sertifikaadi väljaandjale.
3. CN=Apache2204.octox.demo on väljastatava sertifikaadi *common name*.
4. DNS:Apache2204.octox.demo ja DNS:MYWEBSERVER.octox.demo on sertifikaadil olevad SAN DNS nimed, mis peavad kindlasti vastama veebisaidi reaalsele aadressile⁴. Ilmselt pole vaja lisada, et loomulikult peavad need nimed ka nimeserveris lahenema.

³ Lisaks käsureal kirjeldatud sertifikaadi atribuutidele C, O ja CN on võimalik soovi korral lisaks veel kirjeldada atribuudid L, OU ja S. Võib aga kasutada ka ainult CN-i.

⁴ Moodsad brauserid ei pea saiti usaldusväärseks, kui vähemalt üks SAN DNS ei vasta veebisaidi reaalsele aadressile.



Loodud sertifikaadi päringufaili sisu on võimalik vaadata käsuga „openssl req -in Apache2204.csr -noout -text“.

```
uv@ubuntu2204:~/temp$ openssl req -in Apache2204.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: C = EE, O = OctoX, CN = Apache2204.octox.demo
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
      Public-Key: (384 bit)
      pub:
        04:6d:e0:54:c9:00:ad:a4:f5:3f:61:a5:20:07:95:
        43:bf:8d:e3:e6:cd:24:12:b6:52:90:5e:72:0b:fd:
        ea:49:da:2a:a8:2b:23:32:57:d4:96:ac:71:f3:c8:
        b3:05:db:08:b0:d8:55:38:1c:98:11:40:68:35:98:
        88:73:07:62:8f:47:cd:29:8c:ba:d8:4f:a9:80:60:
        5d:99:a1:a3:5a:2c:61:8c:a0:43:67:10:d0:a9:11:
        bf:73:fc:9e:47:4c:12
      ASN1 OID: secp384r1
      NIST CURVE: P-384
  Attributes:
    Requested Extensions:
      X509v3 Subject Alternative Name:
        DNS:Apache2204.octox.demo, DNS:MYWEBSEVER.octox.demo
  Signature Algorithm: ecdsa-with-SHA256
  Signature Value:
    30:65:02:38:4c:a0:90:03:3e:fb:49:ef:ca:68:9f:d5:b2:68:
    31:76:10:f6:d7:9a:87:2b:22:9d:7f:92:9f:f0:ff:b6:f0:26:
    77:8b:ee:96:59:40:3c:52:8a:f3:78:2c:a3:5f:80:fa:02:31:
    00:cd:fd:7b:15:bc:64:36:3f:0c:18:40:bf:b5:0d:37:4c:49:
    16:76:2c:c4:59:58:7d:48:4c:f3:42:1b:d0:f1:5e:1a:40:32:
    3c:b0:c4:ad:3f:01:f7:60:e5:a0:a2:82:1b
```

Pilt 8 - sertifikaadi päringufail sisaldab kahe SAN DNS aadressi kirjutamist sertifikaati

RSA

Juhul, kui mingil põhjusel soovime jätkata RSA algoritmiga, on siia juhendisse jäetud ka vana, üle-eelmise juhendi õpetus RSA sertifikaadipäringu loomiseks. Edasistes punktides selles juhendis aga jätkame eelmises punktis kirjeldatud ECC algoritmil põhineva sertifikaadiga.

Loomesertifikaadi päringu ja privaatvõtme käsuga „openssl req -newkey rsa:2048 -keyout Apache2021.key -sha256 -subj "/CN=Apache5.kaheksa.xi" -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS:Apache2021.kaheksa.xi,DNS:Apache5.kaheksa.xi)") -out Apache2021.csr -nodes“.

```
uv@Ubuntu8:~$ openssl req -newkey rsa:2048 -keyout Apache2021.key -sha256 -subj
"/CN=Apache5.kaheksa.xi" -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(printf
f "[SAN]\nsubjectAltName=DNS:Apache2021.kaheksa.xi,DNS:Apache5.kaheksa.xi")) -ou
t Apache2021.csr -nodes
Generating a RSA private key
.....+++++
+++++
writing new private key to 'Apache2021.key'
-----
uv@Ubuntu8:~$
```

Pilt 9 - privaatvõtme ja sertifikaadi päringu genereerimine

Kollase taustaga märgitud muutujatest:

1. Apache2021.key on sertifikaadi privaatvõti.



2. Apache2021.csr on sertifikaadi päring, mis edastatakse sertifikaadi väljaandjale.
3. Apache5.kaheksa.xi on väljastatava sertifikaadi subjekt.
4. Apache2021.kaheksa.xi ja Apache5.kaheksa.xi on sertifikaadil olevad SAN DNS nimed, mis peab kindlasti vastama veebisaidi aadressile⁵. Ilmselt pole vaja lisada, et loomulikult peavad need nimed ka nimeserveris lahenema.

Loodud sertifikaadi päringufaili sisu on võimalik vaadata käsuga „openssl req -in Apache2021.csr -noout -text“.

```
uv@Ubuntu8: ~
File Edit View Search Terminal Help
uv@Ubuntu8:~$ openssl req -in Apache2021.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: CN = Apache5.kaheksa.xi
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
      RSA Public-Key: (2048 bit)
      Modulus:
        00:c9:4f:a2:54:bd:1a:bb:88:a6:ec:16:c9:3e:28:
        ee:f6:f6:09:3a:d7:e6:8f:a6:7a:4e:57:3b:38:70:
        70:73:b0:01:95:7a:8d:c3:47:46:49:b9:12:52:20:
        08:0c:ed:f5:ec:c5:4e:25:3e:27:9b:98:67:b0:bd:
        c2:cd:00:98:54:36:d4:bf:b8:60:d9:aa:26:de:6a:
        da:11:23:2e:a9:05:94:ff:e8:bb:d2:5e:c2:68:8d:
        63:97:71:5e:0a:a0:49:fc:27:c7:28:c4:7d:53:12:
        1c:e6:2e:9d:bd:81:5b:ff:6a:e5:cf:b5:1a:1b:a3:
        5a:2e:9b:bd:0c:fe:c8:8f:ed:ff:b6:08:9a:1a:69:
        4f:88:a1:1c:c7:9d:84:53:f0:77:2f:db:ba:2a:9a:
        16:f4:78:02:ca:e2:29:f7:f0:f3:61:df:00:ce:3f:
        fa:80:c5:ca:2d:37:a4:2e:a4:8c:be:a2:b3:c9:fd:
        46:4e:20:fb:18:8b:3d:09:6a:be:01:3d:af:29:dd:
        e2:b6:63:3c:3e:46:c1:7a:9b:08:83:c9:32:c5:54:
        b2:e6:3d:a3:68:b6:8d:53:cb:36:c2:20:7d:77:63:
        c7:cf:c9:11:36:b3:47:9b:10:8f:19:66:cb:a4:0f:
        50:f5:35:bf:0d:53:82:cb:ad:3c:1f:5a:1a:2b:70:
        a4:8f
      Exponent: 65537 (0x10001)
  Attributes:
    Requested Extensions:
      X509v3 Subject Alternative Name:
        DNS:Apache2021.kaheksa.xi, DNS:Apache5.kaheksa.xi
      Signature Algorithm: sha256WithRSAEncryption
```

Pilt 10 - sertifikaadi päringufail sisaldab kahe SAN DNS kirjutamist sertifikaati

Sertifikaadi päring

Järgnevalt saadame sertifikaadi päringufaili Apache2204.csr sertifikaadi väljastajale. Meie tingimustes on see testkeskkonna CA. Server väljastab meile sertifikaadi Base-64 kodeeritud formaadis:

⁵ Moodsad brauserid ei pea saiti usaldusväärseks, kui vähemalt üks SAN DNS ei vasta veebisaidi reaalsele aadressile.



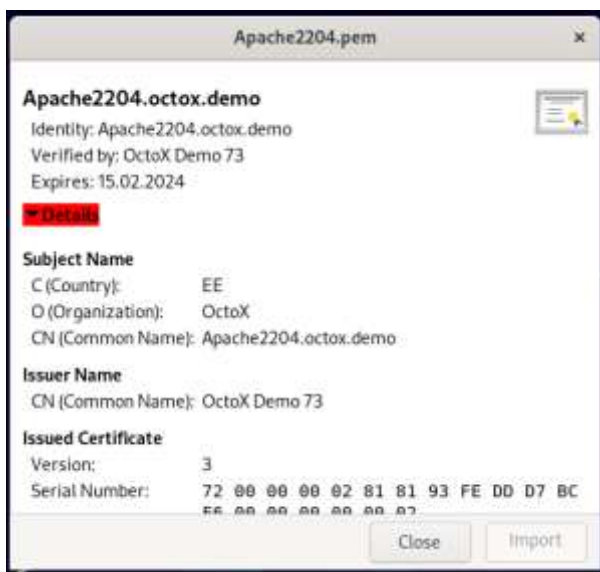
Ubuntu/Apache2 SSL häälestus

Lihtne konfiguratsioonijuhend Eesti EID kaartide vaates



Pilt 11 - sertifikaadi väljanägemine tekstiredaktoris

Avades sertifikaadi Ubuntu's näeme järgmist:



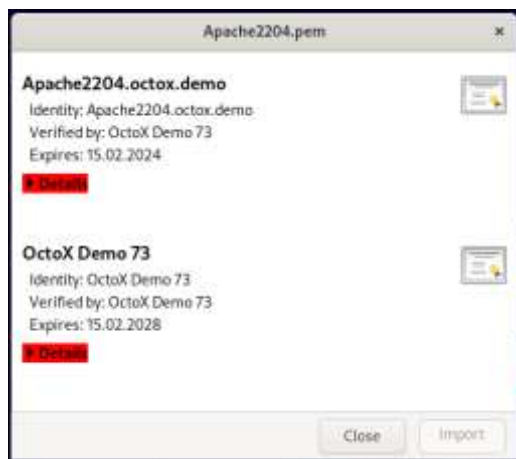
Pilt 12 – ECC sertifikaat Ubuntu's

Sertifikaadis on kirjas ka algoritm ja alternatiivsed subjekti DNS nimed:



Pilt 13 – algoritm ja SAN DNS nimed

Nagu näeme, on sertifikaadi väljastajaks CA nimega „OctoX Demo 73“. Nüüd tuleb meil luua sertifikaadi fail, milles paiknevad nii tulevane veebiserveri TLS sertifikaat kui ka selle väljaandjate ahel. Selleks lisame veebiserveri sertifikaadifailile pem formaadis ka väljastaja sertifikaadi pem formaadis ja salvestame faili nimega Apache2204.pem⁶.



Pilt 14 – veebiserveri sertifikaadiahel Ubuntu

⁶ Eelmises juhendi versioonis kasutasime ka direktiivi SSLCertificateChainFile, kus kirjeldasime veebiserveri sertifikaadi väljastaja(d) pem formaadis. Nüüd loetakse see aga liigseks ja sertifikaadi ja ahela kirjeldamiseks piisab direktiivist SSLCertificateFile. Vt. ka https://httpd.apache.org/docs/2.4/mod/mod_ssl.html#sslcertificatechainfile.



Loodud fail tuleb paigaldada kausta /etc/ssl/certs. Lisaks peame veebiserveri sertifikaadi privaatvõtme paigaldama kausta /etc/ssl/private.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# cp Apache2204.pem /etc/ssl/certs
root@ubuntu2204: /home/uv/temp# cp Apache2204.key /etc/ssl/private
root@ubuntu2204: /home/uv/temp#
```

Pilt 15 - sertifikaadi ja selle privaatvõtme asetamine konteinerisse

Nüüd on Apache2 serveripoolsed sertifikaadi olemas ja korrektselt failisüsteemi paigaldatud.

Virtuaalse veebisaidi loomine

Loome enda konfiguratsioonile eraldiseiva virtuaalse veebisaidi. Esmalt loome kausta /var/www/Apache2204, kuhu paigaldame veebi sisu.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# mkdir /var/www/Apache2204
root@ubuntu2204: /home/uv/temp#
```

Pilt 16 - kausta loomine veebifailidele

Seejärel paigaldame sinna mõne lihtsa ja äratuntava veebilehe. Meie näites võtame testimiseks vaikimisi lehe kaustast /var/www/html/index.html. Oma näites muudame pisut kopeeritud lehe päist ja sisu veendumaks, et veebileht võetakse ikka õigest kohast.

Seejärel teeme valmis virtuaalse saidi konfiguratsioonifaili. Teeme uue faili nimega /etc/apache2/sites-available/Apache2204.conf“ käsuga „nano /etc/apache2/sites-available/Apache2204.conf“.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# nano /etc/apache2/sites-available/Apache2204.conf
root@ubuntu2204: /home/uv/temp#
```

Pilt 17 – teeme uue konfiguratsioonifaili

Nüüd muudame uut konfiguratsioonifaili vastavalt oma soovidele. Lisame sinna järgmise sisu:

```
# Faili algus
```

```
<Virtualhost Apache2204.octox.demo:80>
```

```
# Pöördudes http saidi poole juhatakse meid kahes järgmise rea abil automaatselt https saidile.
```

```
    Servername Apache2204.octox.demo
```

```
    redirect / https://Apache2204.octox.demo
```

```
</Virtualhost>
```

```
<VirtualHost Apache2204.octox.demo:443>
```

Ubuntu/Apache2 SSL häälestus



Lihtne konfiguratsioonijuhend Eesti EID kaartide vaates

Üldinfo

```
ServerName Apache2204.octox.demo:443
DocumentRoot /var/www/Apache2204
```

SSL häälestus

```
SSLEngine on
SSLCertificateFile /etc/ssl/certs/Apache2204.pem
SSLCertificateKeyFile /etc/ssl/private/Apache2204.key
```

Vigade kogumise häälestus

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

</Virtualhost>

Faili lõpp

Aktiveerime loodud konfiguratsiooni käsuga „a2ensite Apache2204.conf“ ja taaskäivitame Apache2 teenuse.



```
root@ubuntu2204:/etc# a2ensite Apache2204.conf
Enabling site Apache2204.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubuntu2204:/etc# systemctl reload apache2
root@ubuntu2204:/etc#
```

Pilt 18 - saidi lubamine ja Apache2 restart

Nüüd saame kasutada ühepoolset SSL-i saidi poole pöördumiseks. Samuti suunatakse meid adressilt <http://Apache2204.octox.demo> aadressile <https://Apache2204.octox.demo>.



Pilt 19 - Apache veebiserver töötab ja kasutab ühepoolset SSL-i!

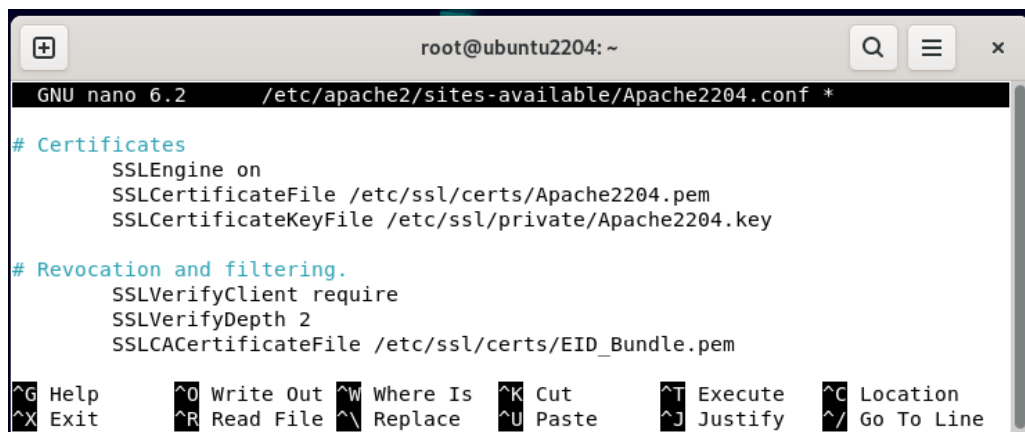
Märkus. Sarnaseid virtuaalseid saite erinevate nimede ja sama IP-aadressiga võime Apache2 veebiserverile luua mitmeid.



Kahepoolse sertifikaadinõude (SSL-i) kehtestamine

Kui soovime, et meie veebisaidile saab ligi end mõne Eesti eID kaardiga autentides, tuleb meil olemasolevat konfiguratsiooni pisut täiendada. Lisame Apache2204.conf failile järgmised read SSL sektsiooni:

- SSLVerifyClient require
- SSLVerifyDepth 2
- SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem



```
root@ubuntu2204: ~
GNU nano 6.2 /etc/apache2/sites-available/Apache2204.conf *
# Certificates
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/Apache2204.pem
  SSLCertificateKeyFile /etc/ssl/private/Apache2204.key
# Revocation and filtering.
  SSLVerifyClient require
  SSLVerifyDepth 2
  SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Pilt 20 - selline on uus konfiguratsioonifaili SSL osa

Nüüd loome uue tekstifaili EID_Bundle.pem⁷, kuhu lisame kõik vajalikud EID juur- ja kesktaseme sertifikaadid Base-64 kodeeritud kujul (EE-GovCA2018, ESTEID2018, EE Certification Centre Root CA, ESTEID-SK 2015). Selle faili abil filtreerime esmalt välja kõik sertimiskeskused, milliste alt väljastatud sertifikaate meie uus veebisait toetab. Meie fail hakkab avatuna Ubuntu nägema välja järgmine:

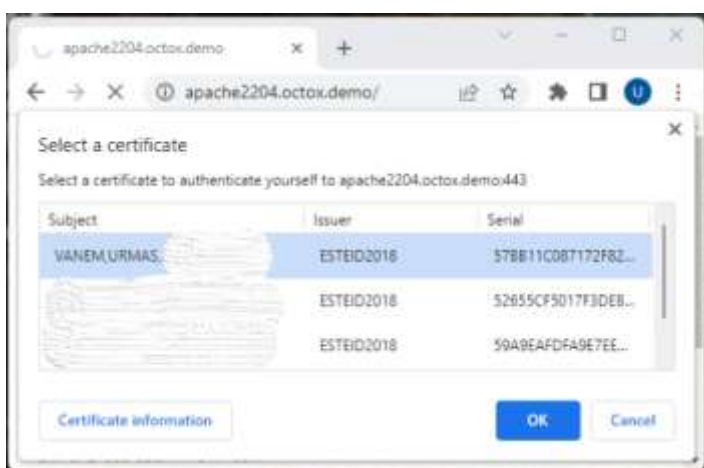
⁷ Allalaetav: https://installer.id.ee/media/id2019/EID_Bundle.pem



Pilt 21 – toetatud sertifikaadid ühes failis koos

Salvestame loodud faili nimega EID_Bundle.pem ja kopeerime selle kausta /etc/ssl/certs. Restardime Apache2 veebiserveri muudatuse jõustumiseks käsuga „systemctl reload apache2“.

Pöördudes peale muudatuse jõustumist uuesti kas veebisaidi Apache2204.octox.demo poole, küsitakse meilt kasutaja sertifikaati.



Pilt 22 - kasutaja sertifikaadi päring



Peale sertifikaadi kinnitamist ja PIN-koodi sisestamist lastakse meid veebisaidile juurde. Kahepoolne SSL töötab.

Apache demokonfiguratsiooni fail selles dokumendis kirjeldatud muutujatega, k.a. osa lisakonfiguratsiooni peatüki all kirjeldatust, on alla laetav aadressilt https://installer.id.ee/media/id2019/Apache_2.4.55_EID_Demo.conf.

Võimalikud lisakonfiguratsioonid

Selle dokumendi eesmärgiks ei ole anda täpseid juhiseid optimaalseks veebisaitide konfigureerimiseks ega turvamiseks. Pigem tahame tutvustada konfiguratsiooni kahepoolse SSL-i kasutamiseks Eesti EID kaartidega. Siiski peame oluliseks pöörata tähelepanu allolevale võimalikule kasulikule informatsioonile.

Tulemüüri reegli loomine, vajadusel

Vaikimisi installatsiooni puhul on Ubuntu tulemüür välja lülitatud. Selle sisse lülitamiseks käivitame terminalis käsu „ufw enable“. Peale tulemüüri sisselülitamist tahame ilmselt luua ka mõned reeglid oma veebiteenusele. Järgnevalt on loetletud Apache puhul järgmised võimalikud variandid:

1. Apache - lubab pordi 80
2. Apache Full – lubab pordid 80 ja 443
3. Apache Secure – lubab pordi 443

Tulemüüri reegli loomiseks tuleb terminalil käivitada käsk „ufw allow 'SOOVITAV REEGEL'“. Näiteks ainult https liikluse lubamiseks tuleb käivitada „ufw allow 'Apache Secure'“.

```
root@ubuntu2204:/home/uv/temp# ufw enable
Firewall is active and enabled on system startup
root@ubuntu2204:/home/uv/temp# ufw allow 'Apache Secure'
Skipping adding existing rule
Skipping adding existing rule (v6)
root@ubuntu2204:/home/uv/temp#
```

Pilt 23 – tulemüüri lubamine ja Apache https reegli loomine

Päring „ufw status“ näitab meile tulemüüri staatust ja olemasolevaid reegleid:

```
root@ubuntu2204:/home/uv/temp# ufw status
Status: active

To Action From
--
Apache Secure ALLOW Anywhere
Apache Secure (v6) ALLOW Anywhere (v6)
root@ubuntu2204:/home/uv/temp#
```

Pilt 24 - tulemüür on aktiivne ja HTTPS on lubatud



OCSP⁸

Garanteeritud OCSP teenus

Vaikimisi lubatakse ülaltoodud konfiguratsiooni rakendades veebisaidile ligi kõik ajaliselt kehtivate sertifikaatidega kasutajad, sertifikaadi tühistusolekut ei kontrollita. Selleks, et kontrollida sertifikaadi kehtivust kasutades SK poolt pakutavat garanteeritud OCSP teenust, tuleb nendega esmalt leping sõlmida. Seejärel antakse tellijale IP-põhine ligipääs OCSP teenusele (aadressiga <http://ocsp.sk.ee>).

Kui ligipääs teenusele on olemas, peame oma Apache2 virtuaalsaidi SSL konfiguratsioonile lisama järgmised read:

- SSLOCSPEnable leaf # – lubame OCSP kehtivuskontrolli kasutaja sertifikaadile.
- SSLOCSPDefaultResponder <http://ocsp.sk.ee> # - määrame OCSP teenuse asukohta.
- SSLOCSPOverrideResponder on # - kasutame keskselt määratud OCSP teenust ka juhul, kui see on kirjas ka kliendi sertifikaadis.
- SSLOCSPResponderCertificateFile /etc/ssl/certs/EID_OCSP.pem – määrame OCSP allkirjastamise sertifikaadi Base-64 kodeeritud formaadis. Täna (21.02.2023) on selleks [SK OCSP RESPONDER 2011](#). (Ja loomulikult tuleb see sertifikaat base-64 formaadis ka konfiguratsioonifailis märgitud asukohta paigutada)

Meie SSL konfiguratsioon näeb nüüd välja järgmine:

```
root@ubuntu2204: -
GNU nano 0.2 /etc/apache2/sites-available/Apache2204.conf *
# Certificates
SSLEngine on
SSLCertificateFile /etc/ssl/certs/Apache2204.pem
SSLCertificateKeyFile /etc/ssl/private/Apache2204.key

# Revocation and filtering.
SSLVerifyClient require
SSLVerifyDepth 2
SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem

# OCSP
SSLLOCSPEnable leaf
SSLLOCSPOverrideResponder on
SSLLOCSPDefaultResponder http://ocsp.sk.ee
SSLOCSPResponderCertificateFile /etc/ssl/certs/EID_OCSP.pem

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line
```

Pilt 25 - SSL konfiguratsioonile on lisatud OCSP osa

Taaskäivitame Apache2 veebiteenuse käivitades terminalis käsu „systemctl reload apache2“. Nüüdsest töötab meil autentimise ajal sertifikaatide kehtivuse kontroll SK garanteeritud OCSP teenuse vastu.

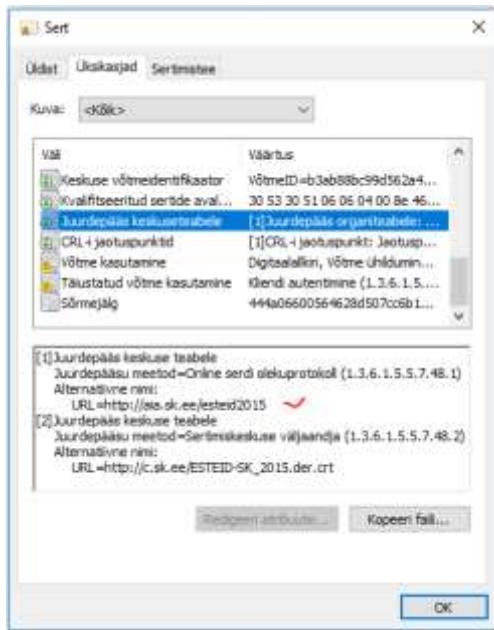
⁸ Sertifikaatide kehtivust on võimalik kontrollida ka sertifikaatide tühistusnimekirjade (CRL) abil, ent sellel me käesolevas dokumendis ei peatu, kuna peame OCSP-põhist lahendust paremaks.



AIA-OCSP teenus

Lisaks garanteeritud (tasulisele) OCSP teenusele pakub SK ka AIA-OCSP teenust, millise puhul sertifikaate kontrollitakse pisut lihtsama OCSP teenuse vastu. Erinevatest ahelatest väljastatud sertifikaatide AIA-OCSP teed on järgmised:

- 1) „ESTEID-SK 2015“ CA tasemelt väljastatud sertifikaadid: <http://aia.sk.ee/esteid2015>
- 2) „ESTEID2018“ CA tasemelt väljastatud sertifikaadid: <http://aia.sk.ee/esteid2018>



Pilt 26 - ESTEID-SK 2015 AIA-OCSP tee sertifikaadis

Lubamaks kasutaja sertifikaadi kontrolli vastu sertifikaadis olevat AIA-OCSP teenust, tuleb meil Apache2 SSL konfiguratsiooni lisada järgmised read:

- SSLOCSPEnable leaf # – lubame OCSP kontrolli kasutaja sertifikaatidele.
- SSLOCSPUseRequestNonce off # - lülitame välja OCSP teenuse vastuse *nonce* nõude.
- # SSLOCSPDefaultResponder # - lisavõimalus serveripoolseks OCSP tee kirjeldamiseks. Nii „ESTEID-SK 2015“ kui „ESTEID2018“ ahelatest väljastatud sertifikaatide kehtivust on võimalik kontrollida mõlema ülalkirjeldatud OCSP tee vastu. Aga kuna sertifikaatidesse on AIA OCSP tee juba sisse kirjutatud, siis siin kirjeldatud konfiguratsiooni puhul ei ole see direktiiv oluline.



```
root@ubuntu2204: ~
GNU nano 6.2 /etc/apache2/sites-available/Apache2204.conf
# Certificates
SSLEngine on
SSLCertificateFile /etc/ssl/certs/Apache2204.pem
SSLCertificateKeyFile /etc/ssl/private/Apache2204.key

# Revocation and filtering.
SSLVerifyClient require
SSLVerifyDepth 2
SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem

# AIA-OCSP
SSLOCSPEnable leaf
SSLOCSPUseRequestNonce off

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line
```

Pilt 27 - SSL konfiguratsioonile on lisatud AIA-OCSP osa

Taaskäivitame Apache2 veebiteenuse käivitades terminalis käsu „systemctl reload apache2“. Nüüdsest töötab meil sertifikaatide kehtivuse kontrolli SK AIA-OCSP teenuse vastu.

Vaikimisi veebisaidi eemaldamine

Apache2 installatsiooniga installeeritakse ka vaikimisi veebisait. Selle eemaldamiseks lahendusest tuleb terminalis käskida „a2dissite 000-default.conf“ ja kinnitada oma otsus käsuga „systemctl reload apache2“.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
root@ubuntu2204: /home/uv/temp# systemctl reload apache2
root@ubuntu2204: /home/uv/temp#
```

Pilt 28 – vaikimisi veebisaidi eemaldamine

Soovituslikud Apache turvasätted

SSL/TLS

Apache versioonil 2.4.55 on vaikimisi lubatud kõik SSL/TLS protokollid, mis on uuemad kui SSL3:

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# grep -l -r "SSLProtocol" /etc/apache2/nods-availa
ble/
/etc/apache2/nods-available/ssl.conf:SSLProtocol all -SSLv3
root@ubuntu2204: /home/uv/temp#
```

Pilt 29 - Apache SSL/TLS vaikimisi konfiguratsioon

Täna sel päeval soovime tungivalt mitte kasutada TLS protokollid versioonist 1.2 madalamaid versioone. Samas on mõnda aega juba kasutusel ka TLS versioon 1.3.

TLS 1.2 on korrektse konfiguratsiooni puhul väga stabiilne ja turvaline, ent täna soovime juba sellest loobumist kasutamaks ainult versiooni 1.3. TLS 1.3 on kiirem ning vaikimisi turvalisem ja nõuab vähem



konfigureerimist. Kui teil ei ole spetsiifilist nõuet TLS 1.2 versiooni lubamiseks, siis soovime edasi minna vaid TLS versiooniga 1.3! Standardlahendustes võiks täna TLS 1.2 olla toetatud vaid tõestatud vajaduse puhul ja sel juhul tuleb olla veendunud, et kasutusel on vaid turvalise šifrikomplektid ja laiendused!

Kui soovime Apache puhul kasutada vaid protokoll TLS 1.3, tuleb konfiguratsioonifaili lisada rida „SSLPROTOCOL -all +TLSv1.3“.

```
SSLPROTOCOL -all +TLSv1.3
```

Pilt 30 - lubame ainsa protokoll TLS 1.3 konfiguratsioonifailis

Toetamiseks TLS versioone 1.2 ja 1.3, tuleb konfiguratsioonireale lisada „+TLSv1.2“.

Alternatiivina saame sama muudatuse teha serveripõhiselt konfigureerides parameetrit SSLPROTOCOL failis /etc/apache2/mods-available/ssl.conf.

Rohkem infot soovitude osas TLS kasutamise kohta võib leida RIA tellitud uuringust [Kruptoalgoritmide ning nende tugi-teekides ja infosüsteemides-2021.pdf](#).

Šifrid (Ciphers)

TLS 1.3 versiooni šifreid peetakse täna kõiki turvaliseks, nii et selle protokoll vaates me turvakaalutlustel lisakonfiguratsiooni looma ei pea.

TLS 1.2 puhul see päris nii ei ole. Apache 2.4.55 versiooniga on vaikimisi kasutusel suur hulk erinevaid TLS šifreid⁹, milliseid näeme käsuga „openssl ciphers -v“.

Vaikimisi on šifrite kasutamise osas defineeritud ainult kaks reeglit:

- 1) HIGH – lubatud on mõned šifrid võtme pikkusega 128 bitti ja kõik tugevamad;
- 2) !aNULL – keelatud on šifrite komplektid mis ei toeta autentimist.

```
SSLCipherSuite HIGH:!aNULL
```

Pilt 31 - serveripõhise konfiguratsiooni kirjeldus failis /etc/apache2/mods-available/ssl.conf

Kui soovime täpsemalt määrata TLS 1.2 protokolliga kasutatavaid šifrite komplekte (ja me ju vajadusel soovime!), saame kasutada Apache kaustapõhises konfiguratsioonifailis käsku SSLCIPHERSUITE. Siin omakorda saame kasutada kas eeldefineeritud muutujaid või täpseid šifrite komplektide kirjeldusi.

Kindlat soovitus erinevate šifrite kasutamiseks ei ole veebisaidile esitatavaid tingimusi teadmata võimalik anda. Küll aga tuleb kindlasti eemaldada loendist ebaturvalised šifrite komplektid. Mõistlik tundub kirjeldada konkreetseid lubatud šifrite komplektid.

Näide:

⁹ Me ei käsitle siin teiste TLS protokollide šifreid kuna eeldame, et versioonist 1.2 vanemad protokollid on keelatud ja 1.3 versiooniga on täna kõik hästi.



- Kasutades järgmist käsurida konfiguratsioonifailis: „SSLCIPHERSUITE 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384'“ – lubatakse vaid kirjeldatud šifrite kasutamine.

Alternatiivina saame kasutatavaid šifreid konfigureerida serveripõhiselt failis /etc/apache2/mods-available/ssl.conf muutes selles parameetrit SSLCIPHERSUITE.

Šifrite osas enama informatsiooni ja soovitude saamiseks suuname teid taas dokumendi [Kruptoalgoritmid-ning-nende-tugi-teekides-ja-infosustemides-2021.pdf](#) juurde!

SSLHONORCIPHERORDER

Oluline šifritega seotud parameeter on ka SSLHONORCIPHERORDER, millise väärtuse konfiguratsioonifailis soovitate keerata ON asendisse. Sel juhul eelistatakse võimalike šifrikombinatsioonide kokkuleppel serveripoolset prioriteeti. Vaikimisi on see määramata ja vaikimisi väärtuseks on määratud „off“.

Kasutajasertifikaatide lisafiltreerimine

Oluline! Kindlustamaks, et meie veebiteenuse poole saavad pöörduda vaid „õiged“ kasutajad korrektsete sertifikaatidega, tuleb serveri konfiguratsioonis kehtestada järgmised nõuded:

- 1) Sertifikaadis korrektse extendedKeyUsage välja olemasolu;
- 2) Sertifikaadi väljastajaks peab olema „ESTEID2018“ või „ESTEID-SK 2015“;
- 3) „ESTEID-SK 2015“ ahelast väljastatud kasutajasertifikaatide DN **ei tohi** sisaldada väärtust O="ESTEID (MOBIIL-ID)".

Selleks tuleb lisada Apache konfiguratsioonile read:

```
<Location "/">
Require expr ( \
    %{SSL_CLIENT_I_DN_CN} == "ESTEID2018" \
and    "TLS Web Client Authentication, E-mail Protection" in PeerExtList('extendedKeyUsage') \
    or %{SSL_CLIENT_I_DN_CN} == "ESTEID-SK 2015" \
and    "TLS Web Client Authentication, E-mail Protection" in PeerExtList('extendedKeyUsage') \
and    %{SSL_Client_S_DN_O} != "ESTEID (MOBIIL-ID)" \
)
</Location>
```

Konfiguratsiooni võib lisada kas virtuaalse hosti või Apache serveri üld-konfiguratsiooni juurde. Seejärel on teenuse poole lubatud pöörduda vaid sertifikaatidega, millistel on korrektne extendedKeyUsage väli ning mis on väljastatud meie poolt lubatud ahelatest ja vastavad ka kõikidele muudele kirjeldatud tingimustele!

Märkuseid!:

- Kui teil on kasutusel mõni muu liikluse filtreerimise vahend/võimalus, siis soovitate turvalise konfiguratsiooni juurutada ka seal. SK ID Solutions (edaspidi SK) on F5 konfiguratsiooni osas publitseerinud järgmise informatsiooni (vt. peakükki „Only accept certificates with trusted key



usage“) artiklis: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>

- SK soovitusel turvaliseks autentimiseks ID-kaardiga on leitavad peatükist „Defence: implement ID-card authentication securely“ juba eelnevalt mainitud artiklist aadressilt: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>
- Soovituslik meetod valesid sertifikaate vältida on kasutada sertifikaatides olevaid OID'e. Paraku ei ole me leidnud täna veel meetodit, kuidas seda serveri tasemel teha. Võimalusel võtke autentimise sertifikaat veebirakenduse tasemel lahti ja kontrollige, kas see sisaldab mõnda korrektseid OID'i. Kui ei sisalda, siis ärge autentige. Täna teadaolevad OID-id on SK publitseerinud peatükis „Only accept certificates with trusted issuance policy“ järgmisel samuti juba eelnevalt mainitud artiklist aadressilt: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>

CA-de sertifikaatide filtreerimine kliendile

Vaikimisi konfiguratsioonis ei piirata kliendi poolel kuvatavate sertifikaatide valikut mis tähendab, et veebiserverisse autentimisel näidatakse kliendile kõiki kasutaja käsutuses olevaid kasutaja autentimise sertifikaate. Meie huvi on kliendi pool näidata aga vaid neid sertifikaate, millised on väljastatud ahelatest „ESTEID2018“ või „ESTEID-SK 2015“. Selleks:

- 1) Loo sertifikaatidele kausta: `mkdir /etc/ssl/certs/client_request_CAs`;
- 2) Paneme sinna „ESTEID2018“ ja „ESTEID-SK 2015“ sertifikaadid Base64 kodeeringus;
- 3) Loo sertifikaatide räsi: „`openssl rehash /etc/ssl/certs/client_request_CAs`“, kausta tekib 2 uut linki/faili milliste nimi vastab räsile:



Pilt 32 – kesktaseme sertifikaadid ja nende räsid

- 4) Lisame Apache SSL häälestuse sektsiooni direktiivi „`SSLCADNRequestPath /etc/ssl/certs/client_request_CAs`“ ja salvestame uue konfiguratsiooni;
- 5) Restardime Apache serveri käsuga: „`systemctl reload apache2`“.

Nüüd saadab Apache server kliendile info, et toetatud on ainult „ESTEID2018“ ja „ESTEID-SK 2015“ ahelatest väljastatud sertifikaadid ja ainult nende CA-de poolt väljastatud sertifikaate kliendi pool ka kuvatakse!

Alternatiivne direktiiv sertifikaatide filtreerimiseks


Direktiivi `SSLCADNRequestPath` võib soovi korral asendada ka direktiiviga `SSLCADNRequestFile`. Sellisel juhul tuleb `SSLCADNRequestFile` direktiivi abil kirjeldada tee failini, mis sisaldab kõikide toetatud



kesktaseme sertimiskeskuste nimekirja PEM formaadis. Näiteks „SSLCADNRequestFile /etc/ssl/certs/SupportedCAs.pem“, kus fail SupportedCAs.pem sisaldab kõiki toetatud kesktaseme sertimiskeskuseid.

HTTP Strict Transport Security (HSTS) lubamine

Lubame mod-headers käsuga „a2enmod headers“ terminalis:



```
root@ubuntu2204:~# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu2204:~# systemctl restart apache2
root@ubuntu2204:~#
```

Pilt 33 – mod-headers lubamine, meie näites HSTS tarbeks

Lisa Apache konfiguratsioonifaili rida: „Header always set Strict-Transport-Security \"max-age=31536000; includeSubDomains\"“.

```
# Enable HSTS.
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

Pilt 34 – HSTS aktiveerimine 30-ks päevaks

Konfiguratsiooni jõustamiseks taaskäivita Apache teenus.

Muud võimalused

Lisaks TLS ja šifrite häälestusele soovime Apache konfiguratsiooni turvalisusele pöörata tähelepanu ka järgmiste punktide vaates:

- Hoida operatsioonisüsteem uuendatuna.
- Hoida Apache uuendatuna.
- Käitle Apache't tavakasutaja õigustes.
- Keela serveri info presenteerimine.
- Eemalda ebaolulised moodulid.
- Lisa ja konfigureeri *Mod Security*.
- Lisa ja konfigureeri *Mod Evasive*.
- Keela *listing* ligipääs vaikimisi kataloogile.
- Luba logimine.
- ...

Palume suhtuda ülaltoodusse kui näidisloendisse demonstreerimaks, mida veel saab Apache turvalisemaks muutmise jaoks ära teha. Põhjalikemaid soovitusi on võimalik leida paljudelt internetilehtedelt: <https://www.google.com/search?q=how+to+secure+apache+server>.