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

Dokumendi info	
Loomise aeg	06.02.2019
Tellija	RIA
Autor	Urmas Vanem, OctoX
Versioon	23.12/2

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 *cipher*'ite soovitused. Uuendatud TLS protokollide kasutamise soovitused. Demokonfi ja dokumendi muutujate nimed on sünkroniseeritud. Muutja: Urmas Vanem
27.01.2021	21.01/3	Lisatud „mobiil-id“ filter. Muutja: Urmas Vanem

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
27.12.2023	23.12/1	Eemaldatud ESTEID-SK 2015 ahel. Muutja: Urmas Vanem
27.12.2023	23.12/2	Eemaldatud vana OCSP responderi sertifikaat. Muutja: Urmas Vanem

Sissejuhatus

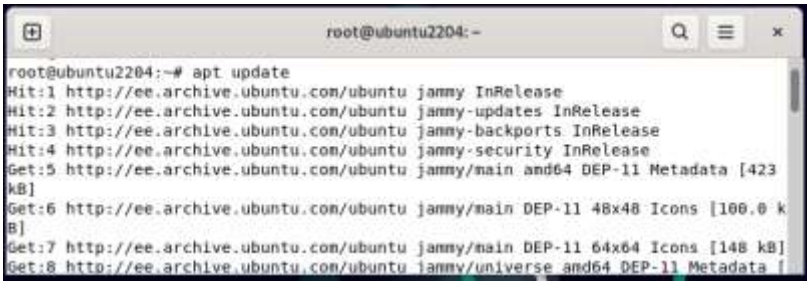
Käesolevas juhendis kirjeldame:

- Kuidas paigaldada 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) veebiserveril!?
- Lisaks vaatame muid võimalusi serveri konfigureerimiseks ja pakume välja ka soovitusi turvalisuse vaates.

Apache2 installatsioon ja häälestus

Installatsioon

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



Pilt 1 – pakkide uuendamine

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

Pilt 2 - Apache2 paigaldus

Eelneva tegevuse tulemusena on Apache server paigaldatud[1]:

[1] Tänane (21.02.2023) viimane Apache versioon on 2.4.55, versioon 2.4.52 on Ubuntuga vaikimisi kaasas.

Pilt 3 - Apache versiooni päring

Uuendame Apache versioonile 2.4.55. Selleks on internetis palju juhiseid olemas[2] ja pikemalt me sellel protseduuril siin ei peatu.

Pilt 4 – Apache versiooniks peale uuendamist on 2.4.55

Apache2 veebiserver versiooniga 2.4.55 töötab nüüd ebaturvalises http režiimis:

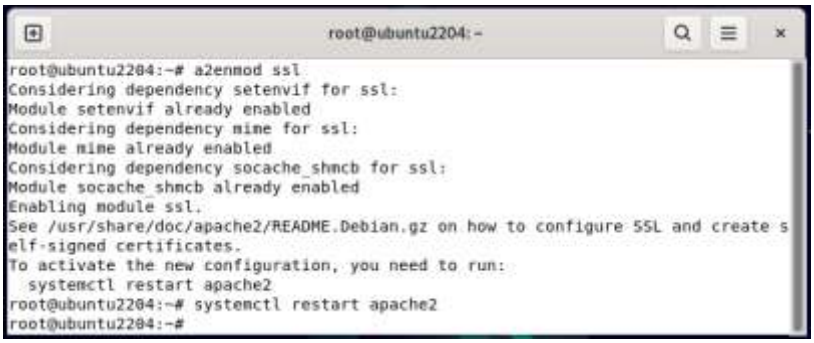
Pilt 5 – Apache veebiserver vaikimisi konfiguratsioonis

Konfiguratsioon

Ühepoolse SSL-i lubamine

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

[2] Näiteks https://ubiq.co/tech-blog/upgrade-apache-version-ubuntu/.



Pilt 6 - SSL lubamine ja teenuse taaskäivitus

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]

Pilt 7 – ECC privaatvõtme ja sertifikaadi päringufaili loomine

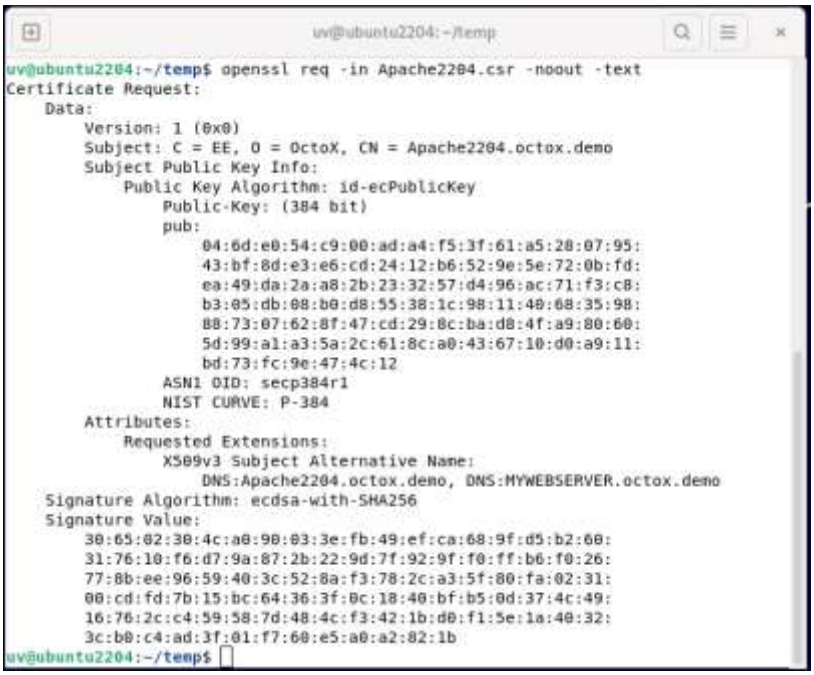
Kollase taustaga muutujatest:

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 aadressile[4]. 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 attribuudid 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“.



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.

Loome sertifikaadi 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“.

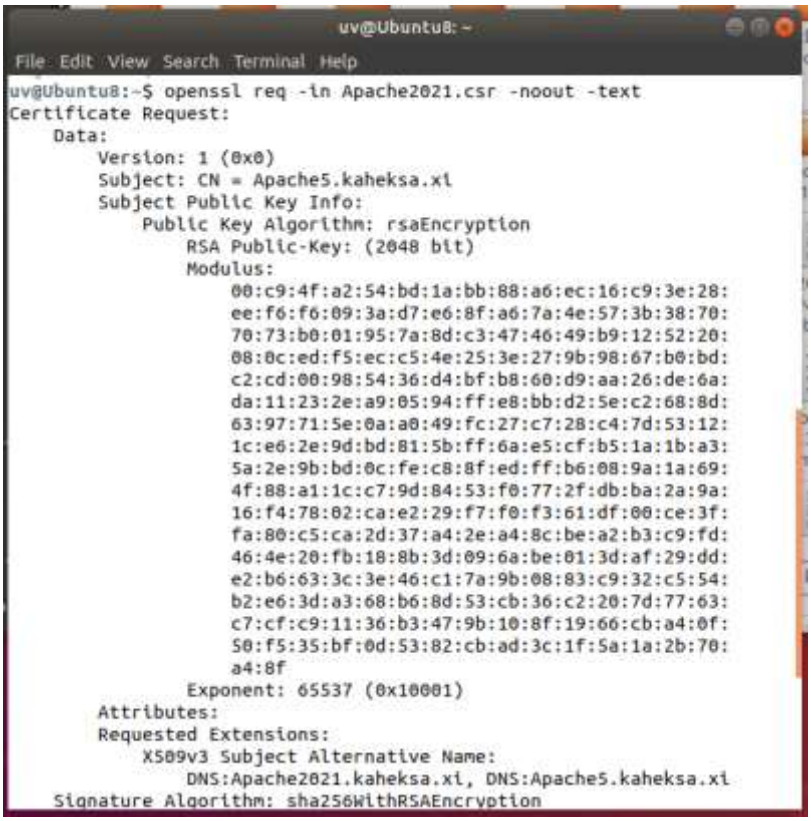
Pilt 9 - privaatvõtme ja sertifikaadi päringu genereerimine

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 aadressile[5]. 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“.



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 Base64 kodeeritud formaadis:

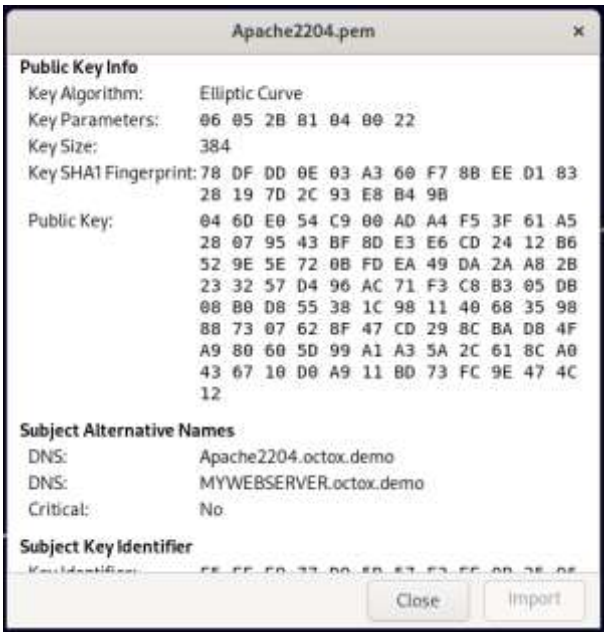
[5] Moodsad brauserid ei pea saiti usaldusväärseks, kui vähemalt üks SAN DNS ei vasta veebisaidi reaalsele aadressile.

Pilt 11 - sertifikaadi väljanägemine tekstiredaktoris

Avades sertifikaadi Ubuntus näeme järgmist:

Pilt 12 – ECC sertifikaat Ubuntus

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.pem.

Pilt 14 – veebiserveri sertifikaadiahel Ubuntus

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

Pilt 15 - sertifikaadi ja selle privaatvõtme asetamine konteinerisse

Nüüd on Apache2 serveripoolsed sertifikaadi olemas ja korrektselt failisüsteemi paigaldatud.

Virtuaalse veebisaidi loomine

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

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“.

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 juhitakse meid kahes järgmise rea abil automaatselt https saidile.
        Servername Apache2204.octox.demo
        redirect / https://Apache2204.octox.demo
</Virtualhost>

<VirtualHost Apache2204.octox.demo:443>

# Üldinfo
        ServerName Apache2204.octox.demo:443
```

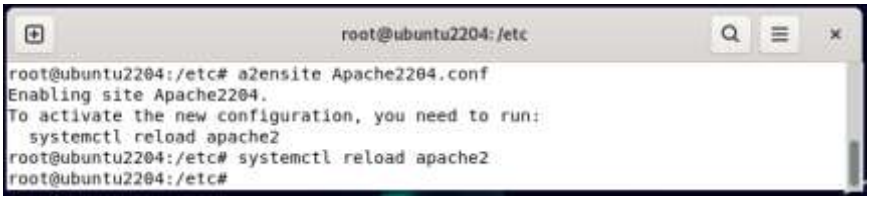
```
        DocumentRoot /var/www/Apache2204

# 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.



Pilt 18 - saidi lubamine ja Apache2 taaskäivitus

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

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

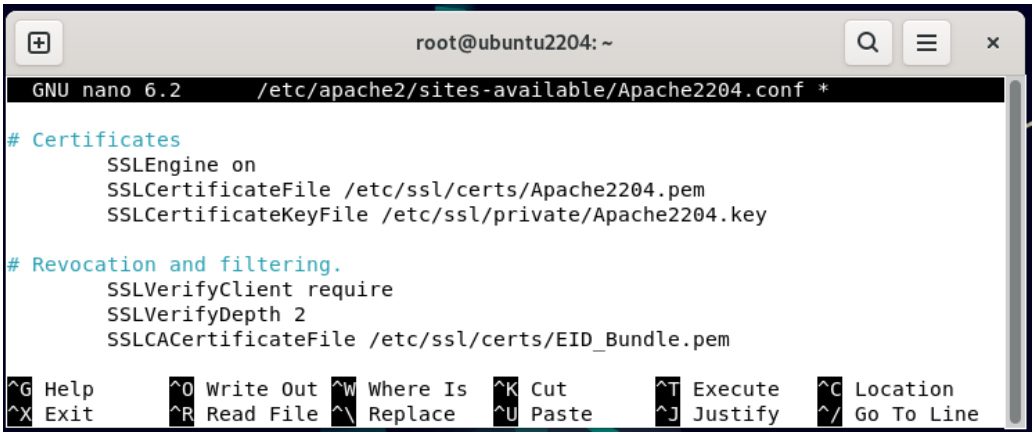
Märkus. Sarnaseid virtuaalseid saite erinevate nimede ja sama IP-aadressiga võime Apache2 veebiserverile luua mitmeid.

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:

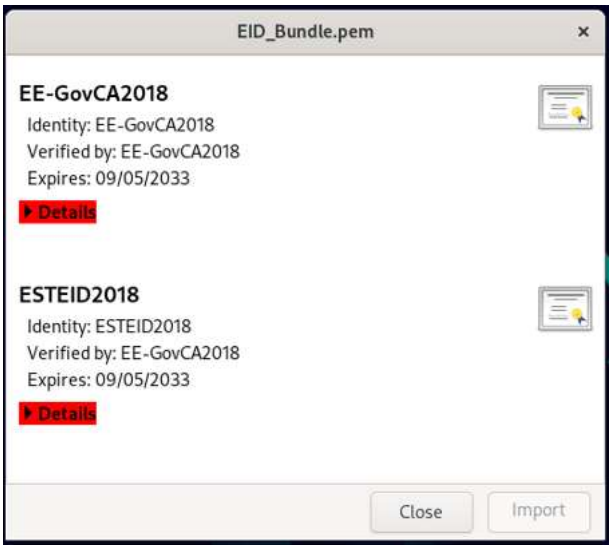
- SSLVerifyClient require

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



Pilt 20 - selline on uus konfiguratsioonifaili SSL osa

Nüüd loome uue tekstifaili EID_Bundle.pem[6], kuhu lisame eID juur- ja kesktaseme sertifikaadid Base64 kodeeritud kujul (EE-GovCA2018, ESTEID2018). Selle faili abil filtreerime esmalt välja kõik sertimiskeskused, mille alt väljastatud sertifikaate meie uus veebisait toetab. Meie fail hakkab avatuna Ubuntus nägema välja järgmine:

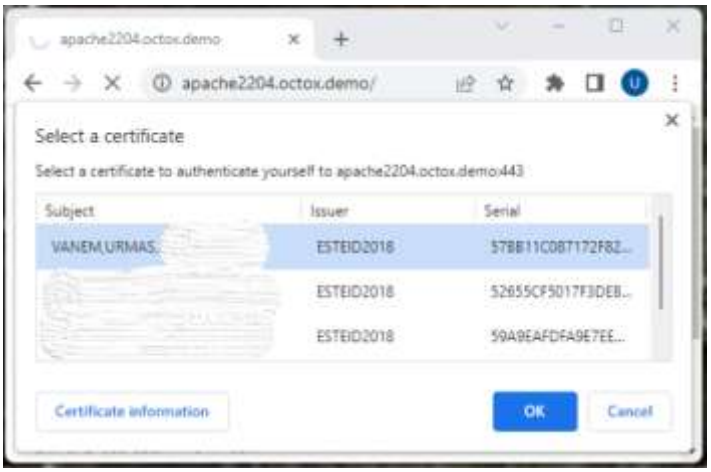


Pilt 21 – toetatud sertifikaadid ühes failis koos

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

[6] Allalaetav: https://installer.id.ee/media/id2019/EID_Bundle.pem

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

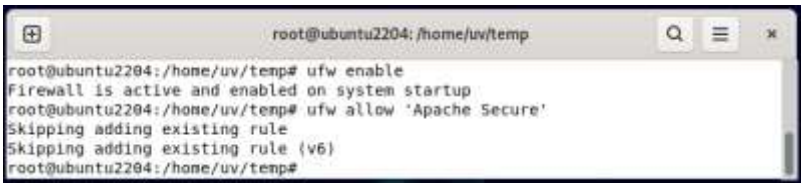
Selle dokumendi eesmärgiks ei ole anda täpseid juhiseid optimaalseks veebisaitide konfigureerimiseks ega turvamiseks. Pigem tahame tutvustada konfiguratsiooni kahepoolse SSL-i kasutamiseks Eesti eID kaartidega. Siiski peame oluliseks pöörata tähelepanu allolevale kasulikule informatsioonile.

Tulemüüri reegli loomine, vajadusel

Vaikimisi installatsiooni puhul on Ubuntul 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’“.



Pilt 23 – tulemüüri lubamine ja Apache https reegli loomine

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

Pilt 24 - tulemüür on aktiivne ja HTTPS on lubatud

OCSP[7]

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 lubatakse tellijale ligipääs OCSP teenusele (aadressiga http://ocsp.sk.ee) kas sertifikaadi- või IP põhiselt.

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 asukoha.
- SSLOCSPOverrideResponder on # - kasutame keskselt määratud OCSP teenust ka juhul, kui see on kirjas ka kliendi sertifikaadis.

Meie SSL konfiguratsioon näeb nüüd välja järgmine:

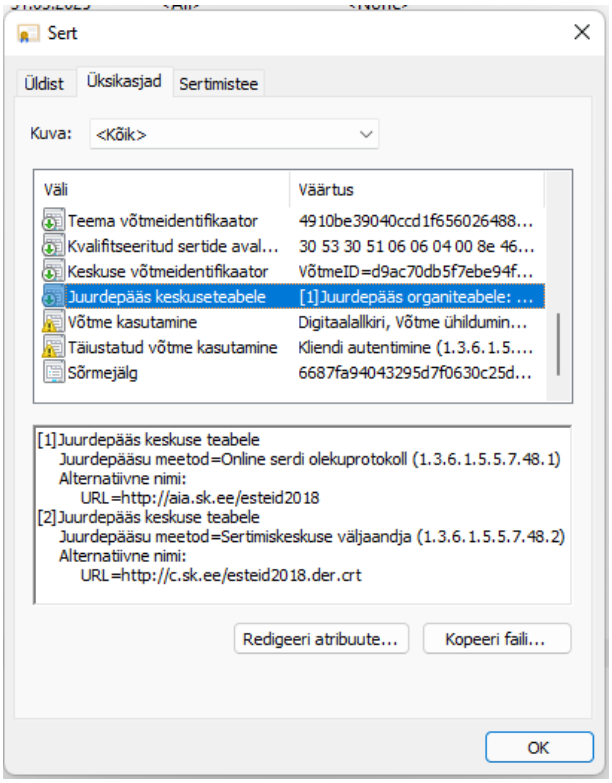
[7] Sertifikaatide kehtivust on võimalik kontrollida ka sertifikaatide tühistusnimekirjade (CRL) abil, ent sellel me käesolevas dokumendis ei peatu, kuna peame OCSP-põhist lahendust paremaks.

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.

AIA OCSP teenus

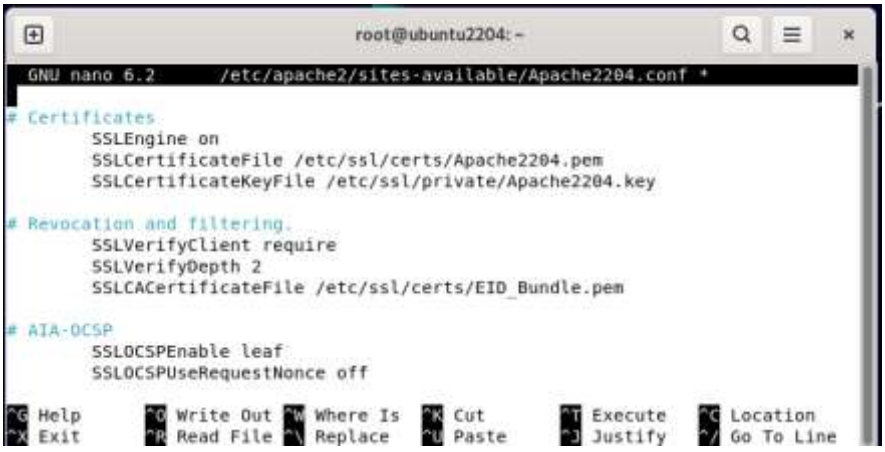
Lisaks garanteeritud (tasulisele) OCSP teenusele pakub SK ka vaba ligipääsuga AIA OCSP teenust, mille puhul sertifikaate kontrollitakse pisut lihtsama OCSP teenuse vastu. eID 2018 sertifikaatide AIA OCSP tee on http://aia.sk.ee/esteid2018.



Pilt 26 – ESTEID2018 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.

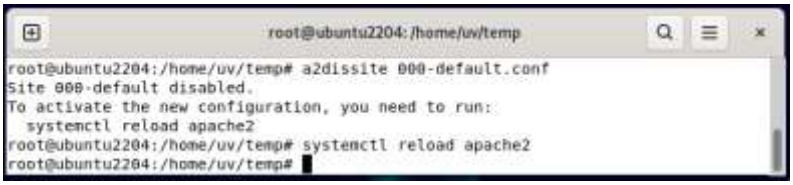


Pilt 27 - SSL konfiguratsioonile on lisatud AIA OCSP osa

Taaskäivitame Apache2 veebiteenuse käivitades terminalis käsu „systemctl reload apache2“. Nüüdsest töötab meil sertifikaatide kehtivuse kontroll 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“.

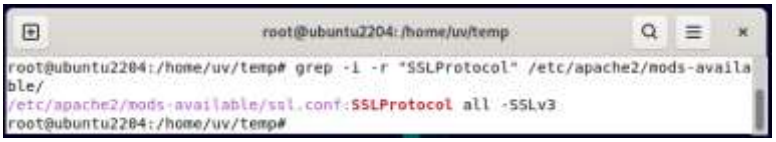


Pilt 28 – vaikimisi veebisaidi eemaldamine

Soovituslikud Apache turvasätted

SSL/TLS

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



Pilt 29 - Apache SSL/TLS vaikimisi konfiguratsioon

Tänapäeval soovitame tungivalt mitte kasutada TLS protokolli 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 soovitame 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 soovitame 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 turvalised 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 protokolli TLS 1.3 konfiguratsioonifailis

Toetamaks 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 TLS protokolli kasutamise soovituste kohta leiab RIA tellitud krüptograafiliste algoritmide elutsükli uuringust aadressil https://www.id.ee/artikkel/kruptograafiliste-algoritmide-elutsukli-uuringud-2/.

Šifrikomplektid (Cipher suites)

TLS 1.3 versiooni šifreid peetakse täna kõiki turvaliseks, nii et selle protokolli 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 šifreid[8], mida 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 šifrikomplekte (ja me ju vajadusel soovime!), saame kasutada Apache kaustapõhises konfiguratsioonifailis käsku SSLCIPHERSUITE. Siin omakorda saame kasutada kas eeldefineeritud muutujaid või täpseid šifrikomplektide kirjeldusi.

Kindlat soovitust erinevate šifrikomplektide kasutamiseks ei ole veebisaidile esitatavaid tingimusi teadmata võimalik anda. Küll aga tuleb kindlasti eemaldada loendist ebaturvalised šifrikomplektid. Mõistlik tundub kirjeldada konkreetsed lubatud šifrikomplektid.

Näide:

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

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

Rohkem infot šifrikomplektide soovituste kohta leiab RIA tellitud krüptograafiliste algoritmide elutsükli uuringust aadressil https://www.id.ee/artikkel/kruptograafiliste-algoritmide-elutsukli-uuringud-2/.

SSLHONORCIPHERORDER

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

[8] 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.

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.

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') \
)
</Location>
```

Konfiguratsiooni võib lisada kas virtuaalse hosti või Apache serveri üld-konfiguratsiooni juurde. Seejärel on teenuse poole lubatud pöörduda vaid sertifikaatidega millel on korrektne extendedKeyUsage väli ning mis on väljastatud meie poolt lubatud ahelast!

Märkuseid!:

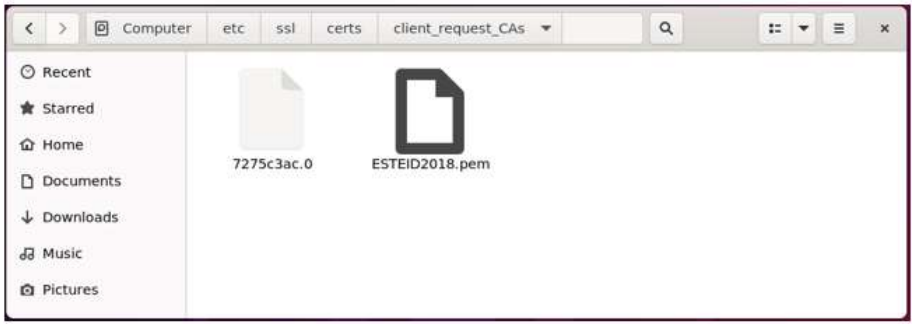
- Kui teil on kasutusel mõni muu liikluse filtreerimise vahend/võimalus, siis soovitame turvalise konfiguratsiooni juurutada ka seal. SK ID Solutions (edaspidi SK) on F5 konfiguratsiooni osas publitseerinud järgmise informatsiooni (vt. peakükki „Only accept certificates with trusted key usage“) artiklis: https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide
- SK soovitused 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 korrektsed 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 aadressil: 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, mis on väljastatud ahelast ESTEID2018. Selleks:

1) loome sertifikaatidele kausta: mkdir /etc/ssl/certs/client_request_CAs;
2) paneme sinna ESTEID2018 sertifikaadi Base64 kodeeringus;

3) loome sertifikaatide räsi: „openssl rehash /etc/ssl/certs/client_request_CAs“, kausta tekib 1 uus link/fail:



Pilt 32 – kesktaseme sertifikaat ja selle räsi

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

Nüüd saadab Apache server kliendile info, et toetatud on ainult ESTEID2018 ahelast väljastatud sertifikaadid ja ainult selle 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:

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 šifrikomplektide häälestusele soovitame 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.