



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

Dokumendi info	
Loomise aeg	08.02.2019
Tellijä	RIA
Autor	Urmas Vanem, OctoX
Versioon	23.12/1

Versiooni info		
Kuupäev	Versioon	Muutused/märkused
08.02.2019	19.02/1	Avalik versioon.
28.02.2019	19.02/2	Märkused kliendi sertifikaatide kehtivuse kontrolli kohta. Vaikimisi veebilehe eemaldamine. Muutja: Urmas Vanem
12.12.2019	19.12/1	Lisatud Nginx soovituslikud turvasätted. Muutja: Urmas Vanem
30.12.2020	20.12/1	Nginx uuendatud versioonile 20.04.1. NGINX uuendatud versioonile 1.19.6. Muudetud konfiguratsiooni haldamine (sites-... -> conf.d). Lisatud OCSP põhiste tühistusnimekirjade kasutamise võimalus, soovituslikud turvasätted ja „valede“ CA-de sertifikaatide blokeerimise kirjeldus. Muutja: Urmas Vanem
13.01.2021	21.01/1	Lisatud demo-konfiguratsiooni fail. Lisatud HSTS konfiguratsioon. Muutja: Urmas Vanem
25.01.2021	21.01/2	Muudetud on HSTS soovitusi. Muudetud on SSL/TLS ja šifrite kasutamise soovitusi. Lisatud on mõned lisaturvalisuse tõstmise soovitusid. Muutja: Urmas Vanem
28.04.2021	21.04/1	Eemaldatud aegunud ESTEID-SK 2011 sertifikaatide tugi. Muutja: Urmas Vanem
25.11.2021	21.11/1	Muudetud Ubuntu platvorm versioonile Ubuntu Server 21.10. Uuendatud NGINX platvorm versioonile 1.21.4. Lisatud on ECC sertifikaatide loomine veebiserveril. Täiendatud on TLS ja Cipher soovitusi. Muutja: Urmas Vanem



22.02.2023	23.02/1	Ubuntu on uuendatud versioonile Ubuntu Server 22.04 ja NGINX versioonile 1.23.3. Uuendatud on ka virtuaalhosti konfiguratsiooni. Muutja: Urmas Vanem
18.12.2023	23.12/1	Eemaldatud ESTEID-SK 2015 ahel. Muutja: Urmas Vanem

Sissejuhatavalt

Käesolevas juhendis kirjeldame:

- Kuidas paigaldada ja häälestada Nginx 1.23.3 veebiserver Ubuntu 22.04 serveril!?
- Kuidas häälestada HTTPS (ühepoolne SSL) veebiserveril!?
- Kuidas häälestada eID kaartidega autentimine (kahepoolne SSL) veebiserveril?
- Kuidas häälestada OCSP kontroll nii vastu garanteeritud kui AIA OCSP teenust?
- Kuidas turvata oma veebiserverit?

Lisaks vaatame muid konfiguratsioonivõimalusi, nagu kuidas HTTP liiklus suunata HTTPS kanalisse jpm.

Nginx installatsioon ja häälestus

Installatsioon

Ubuntu 22.04 versiooni puhul installeeritakse vaikes juhiste puhul NGINX versioon 1.18. Kuna aga soovime oma demojuhendis kasutada viimast versiooni 1.23.3, mille võimalusi on oluliselt täiendatud (lisatud on ka OCSP kasutamise võimalus alates versioonist 1.19), siis tuleb enne installatsiooni teha täiendavaid muudatusi.

1.23.3 versiooni installatsiooniks Ubuntu 22.04 versiooni alla tuleb teha järgmised tegemised (veebruaris 2023):

1. Terminalis käivitada käsk: „wget https://nginx.org/keys/nginx_signing.key“.

```
root@ubuntu2204:~# wget https://nginx.org/keys/nginx_signing.key
--2023-02-22 08:08:31-- https://nginx.org/keys/nginx_signing.key
Resolving nginx.org (nginx.org)... 2a05:d014:edb:5704::6, 2a05:d014:edb:5702::6,
3.125.197.172, ...
Connecting to nginx.org (nginx.org)|2a05:d014:edb:5704::6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1561 (1.5K) [application/octet-stream]
Saving to: 'nginx_signing.key'

nginx_signing.key  100%[=====] 1.52K  --.-KB/s  in 0s

2023-02-22 08:08:31 (216 MB/s) - 'nginx_signing.key' saved [1561/1561]

root@ubuntu2204:~#
```

Pilt 1 – NGINX võtme hankimine



- Terminalis käivitada võtme lisamiseks käsk: „apt-key add nginx_signing.key“.

```
root@ubuntu2204: ~  
root@ubuntu2204:~# apt-key add nginx_signing.key  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
root@ubuntu2204:~#
```

Pilt 2 - võtme lisamine

- Lisada faili /etc/apt/sources.list lõppu read (avame faili käsuga „nano /etc/apt/sources.list“):
deb https://nginx.org/packages/mainline/ubuntu/ jammy nginx
deb-src https://nginx.org/packages/mainline/ubuntu/ jammy nginx

```
root@ubuntu2204: ~  
GNU nano 6.2 /etc/apt/sources.list *  
# deb-src http://ee.archive.ubuntu.com/ubuntu jammy-security multiverse  
  
deb [arch=amd64] http://nginx.org/packages/mainline/ubuntu/ jammy nginx  
deb-src http://nginx.org/packages/mainline/ubuntu/ jammy nginx  
  
^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 3 – lähtefailide täiendamine

- Vajadusel eemalda muud nginx versioonid käsuga „apt-get remove nginx-common“.
- Terminalis käivitada käsk: „apt-get update“.
- Terminalis käivitada käsk: „apt-get install nginx“.

```
root@ubuntu2204: ~  
root@ubuntu2204:~# apt-get install nginx  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  nginx  
0 upgraded, 1 newly installed, 0 to remove and 78 not upgraded.  
Need to get 1,011 kB of archives.  
After this operation, 3,281 kB of additional disk space will be used.  
Get:1 http://nginx.org/packages/mainline/ubuntu jammy/nginx amd64 nginx amd64 1.23.3-1~jammy [1,011 kB]  
Fetched 1,011 kB in 0s (3,002 kB/s)
```

Pilt 4 – NGINX versiooni 1.23.3 installatsioon

Installatsiooni aknast näeme, et installeeritakse meie soovitud versioon 23.3-1~jammy. NGINX versiooni saame kontrollida ka käsuga „nginx -v“:



```
root@ubuntu2204: ~  
root@ubuntu2204:~# nginx -v  
nginx version: nginx/1.23.3  
root@ubuntu2204:~#
```

Pilt 5 - NGINX versiooni kontroll

Kuna me ei plaani hakata kasutama vaikimisi konfiguratsiooni(faili) (ja me ju ei plaani), siis keelame ka selle käsuga: „mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.conf.disabled“.

Konfiguratsioon

Ühepoolse SSL-i kuvamine

Sertifikaadi privaatvõtme ja päringufaili loomine

ECC

Loomes esmalt ECC algoritmil baseeruva privaatvõtme:

1. „openssl ecparam -name secp384r1 -genkey -noout -out **nginx123.key**“

```
root@ubuntu2204: /home/uv/temp  
root@ubuntu2204:/home/uv/temp# openssl ecparam -name secp384r1 -genkey -noout -o  
ut nginx123.key  
root@ubuntu2204:/home/uv/temp# ls  
nginx123.key  
root@ubuntu2204:/home/uv/temp#
```

Pilt 6 - privaatvõtme loomine

Ja seejärel loome privaatvõtme baasil sertifikaadi päringufaili CA serverile:

2. „openssl req -new -key **nginx123.key** -out **nginx123.csr** -subj /C=EE/O=OctoX/CN=**nginx123.octox.demo** -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS:**nginx123.octox.demo**,DNS:**MYWEBSERVER.octox.demo**"))“¹

```
root@ubuntu2204: /home/uv/temp  
root@ubuntu2204:/home/uv/temp# openssl req -new -key nginx123.key -out nginx123.  
csr -subj /C=EE/O=OctoX/CN=nginx123.octox.demo -reqexts SAN -config <(cat /etc/s  
sl/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS:nginx123.octox.demo,DNS:MYWEB  
SERVER.octox.demo"))  
root@ubuntu2204:/home/uv/temp# ls  
nginx123.csr nginx123.key  
root@ubuntu2204:/home/uv/temp#
```

Pilt 7 – sertifikaadi päringufaili loomine

¹ Lisaks käsueal kirjeldatud sertifikaadi informatiivsetele parameetritele C, O ja CN on võimalik soovi korral lisaks veel kirjeldada atribuudid L, OU ja S. Võib kasutada ka ainult CN-i.



Kollase taustaga muutujatest:

1. nginx123.key on sertifikaadi privaatvõti;
2. nginx123.csr on sertifikaadi päringufail, mis edastatakse sertifikaadi väljaandjale;
3. CN=nginx123.octox.demo on väljastatava sertifikaadi *common name*;
4. DNS:nginx123.octox.demo ja DNS:MYWEBSERVER.octox.demo on sertifikaadil olevad SAN DNS nimed, mis peab kindlasti vastama veebisaidi aadressile². Ilmselt pole vaja lisada, et loomulikult peavad need nimed ka nimeserveris lahenema.

Loodud sertifikaadi päringufaili sisu on võimalik vaadata käsuga „openssl req -in nginx123.csr -noout -text“.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204:/home/uv/temp# openssl req -in nginx123.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: C = EE, O = OctoX, CN = nginx123.octox.demo
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (384 bit)
    pub:
      04:93:d2:e2:3a:8a:e6:12:01:c7:bb:53:11:b1:7f:
      84:e0:4c:03:1e:9d:26:ab:aa:f1:38:aa:a1:ca:ba:
      b2:ea:6c:31:1d:60:dc:3b:6e:21:62:01:c9:1c:f0:
      d4:56:3a:3d:e3:31:79:75:f2:7d:b6:48:89:a4:5e:
      21:a8:64:ad:c3:12:5f:04:31:7b:a0:e9:29:e8:a6:
      c4:87:5c:ee:fe:dc:a9:b9:34:89:e3:5b:85:1f:41:
      7b:c7:3d:18:01:d8:b0
    ASN1 OID: secp384r1
    NIST CURVE: P-384
  Attributes:
    Requested Extensions:
      X509v3 Subject Alternative Name:
        DNS:nginx123.octox.demo, DNS:MYWEBSERVER.octox.demo
  Signature Algorithm: ecdsa-with-SHA256
  Signature Value:
    30:66:02:31:00:dc:77:49:17:12:6a:58:05:dc:90:c7:d4:d0:
    4d:be:f0:e1:10:ec:2a:c5:24:31:7f:5b:e2:cc:90:35:57:f9:
    53:41:a7:08:01:f4:a9:2d:94:10:53:ba:b1:c9:22:7d:70:02:
    31:00:f3:33:94:6e:64:1a:75:d5:6e:e1:58:21:a9:23:24:31:
    a5:75:45:36:19:ae:64:37:41:d3:2d:49:ab:c4:d9:75:ee:41:
    74:80:be:be:68:e7:4f:49:f7:4a:7f:fc:e7:03
root@ubuntu2204:/home/uv/temp#
```

Pilt 8 - sertifikaadi päringufail sisaldab kahe SAN DNS kirjutamist sertifikaati

RSA

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

² Moodsad brauserid ei pea saiti usaldusväärseks, kui vähemalt üks SAN DNS ei vasta veebisaidi reaalsele aadressile.



Loome sertifikaadi päringu ja privaatvõtme käsuga „openssl req -newkey rsa:2048 -keyout NGINX20PRIV.key -sha256 -subj "/CN=Nginx20.kaheksa.xi" -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS: Nginx20.kaheksa.xi,DNS: Nginx22.kaheksa.xi ") -out NGINX20.csr -nodes“.

```
root@Ubuntu2020:/home/uv# openssl req -newkey rsa:2048 -keyout NGINX20PRIV.key -
sha256 -subj "/CN=Nginx20.kaheksa.xi" -reqexts SAN -config <(cat /etc/ssl/openssl
.cnf <(printf "[SAN]\nsubjectAltName=DNS: Nginx20.kaheksa.xi,DNS: Nginx22.kahek
sa.xi ") -out NGINX20.csr -nodes
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'NGINX20PRIV.key'
-----
```

Pilt 9 - privaatvõtme ja sertifikaadi päringu genereerimine

Kollase taustaga märgitud muutujatest:

1. NGINX20PRIV.key on sertifikaadi privaatvõti;
2. NGINX20.csr on sertifikaadi päring, mis edastatakse sertifikaadi väljaandjale;
3. Nginx20.kaheksa.xi on väljastatava sertifikaadi subjekt;
4. Nginx20.kaheksa.xi ja Nginx22.kaheksa.xi on sertifikaadil olevad SAN DNS nimed, mis peab kindlasti vastama veebisaidi aadressile³. Ilmselt pole vaja lisada, et loomulikult peavad need nimed ka nimeserveris lahenema ja et piisab ka ühest nimest.

Loodud sertifikaadi päringufaili sisu on võimalik vaadata käsuga „openssl req -in NGINX20.csr -noout -text“.

³ Veebibrauserid ei pea saiti usaldusväärseks, kui vähemalt üks SAN DNS ei vasta veebisaidi reaalsele aadressile.

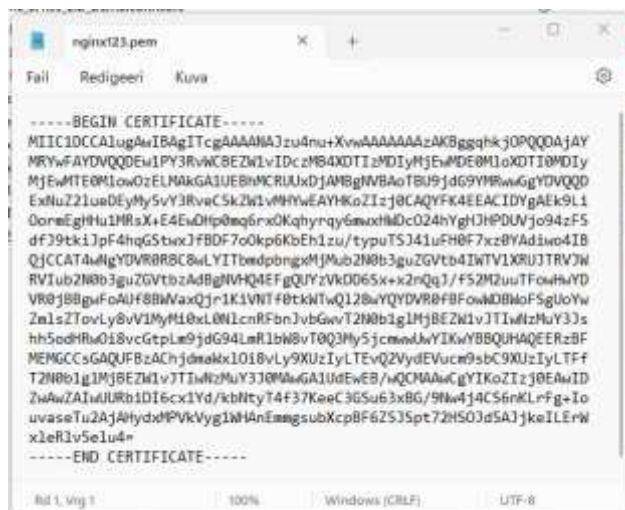


```
root@Ubuntu2020:/home/uv# openssl req -in NGINX20.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: CN = Nginx20.kaheksa.xi
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
      RSA Public-Key: (2048 bit)
      Modulus:
        00:f1:62:c3:ed:1d:0b:ea:cb:7c:22:17:41:1e:e3:
        c1:02:a6:7b:f3:72:13:ae:8d:72:72:6f:09:77:d6:
        51:84:4b:2a:f6:7b:65:9d:9f:f3:2a:0c:16:e5:26:
        47:70:aa:3e:c8:4c:50:62:5b:6c:2a:49:ea:51:01:
        60:5c:94:2c:d6:1d:78:70:eb:41:88:6c:09:c8:2f:
        e4:d5:bb:2f:fb:ec:2f:9d:0c:42:66:b5:de:91:e3:
        60:62:ff:94:11:21:aa:de:bb:52:bd:20:a6:ff:b4:
        c3:92:0a:5b:b5:fc:2f:88:bc:44:3e:b4:5b:a4:ec:
        de:49:16:b6:c0:13:ed:d0:e2:ee:d0:58:bc:cb:36:
        32:c9:1b:6d:8f:79:db:83:22:fd:fe:a7:9a:b2:cd:
        26:b1:d7:52:c4:0c:40:6d:0e:49:b5:18:07:c2:3c:
        c0:c9:70:5d:06:da:0a:e6:01:1a:a4:78:19:aa:a7:
        38:1c:9d:36:07:4d:db:d2:b5:7b:50:f1:4b:d0:c7:
        5d:90:86:92:2d:a6:ea:d7:d2:09:8f:51:e8:b6:52:
        07:b1:1e:5e:ca:65:f3:d4:69:52:f1:d9:47:02:24:
        98:42:70:83:bc:49:13:c1:92:51:f7:ca:b2:fa:f6:
        a7:08:13:c1:74:23:d6:58:ab:27:d5:e5:02:20:3f:
        11:3b
      Exponent: 65537 (0x10001)
  Attributes:
    Requested Extensions:
      X509v3 Subject Alternative Name:
        DNS:Nginx20.kaheksa.xi, DNS:Nginx22.kaheksa.xi
  Signature Algorithm: sha256WithRSAEncryption
  5c:ad:79:7d:be:e1:e0:02:a5:26:11:73:76:b0:77:63:5a:47:
```

Pilt 10 - sertifikaadi päringufail sisaldab kahe SAN DNS kirjutamist sertifikaati

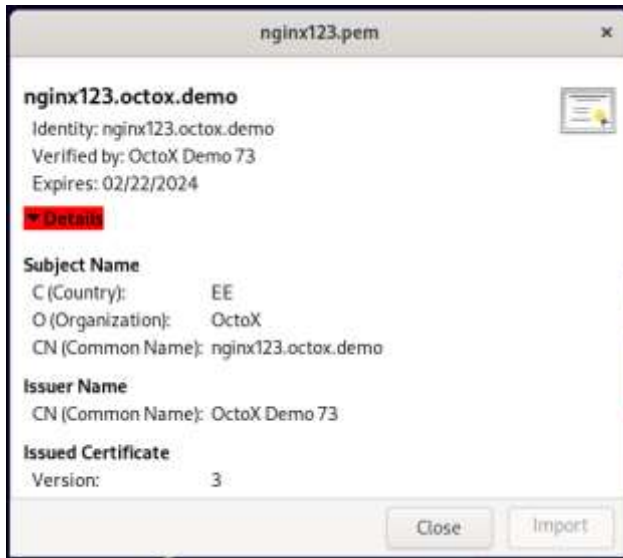
Sertifikaadi päring

Järgnevalt saadame sertifikaadi päringufaili nginx123.csr sertifikaadi väljastajale. Meie tingimustes on see testkeskkonna CA. Server väljastab meile sertifikaadi Base64 kodeeritud formaadis.



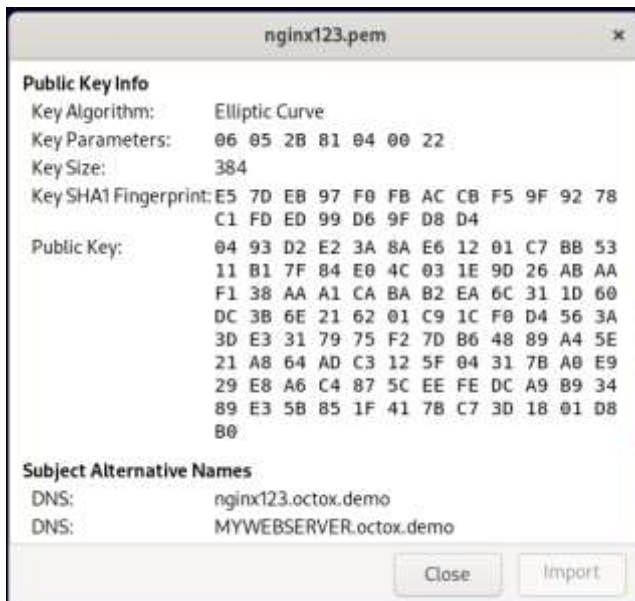
Pilt 11 - sertifikaat pem-formaadis tekstiredaktoris

Avades sertifikaadi Ubuntu failihalduris näeme järgmist:



Pilt 12 - sertifikaat Ubuntu

Sertifikaadis on kirjas ka alternatiivsed subjekti DNS nimed:



Pilt 13 – algoritm ja SAN DNS nimed

Salvestame saadud sertifikaadi kasutaja töökausta nimega nginx123.pem.

Nagu näeme, on sertifikaadi väljastajaks CA nimega „OctoX Demo 73“. Nüüd peame hankima endale väljastaja CA sertifikaadi Base64 kodeeringus ja salvestama selle kasutaja kodukausta nimega OD73.pem.

Paneme kõik ühepoolseks SSL-iks kasutatavad sertifikaadid kokku ühte faili, kusjuures esimene sertifikaat selles failis peab olema veebiserveri sertifikaat. Meie näitel peame faili kokku panema



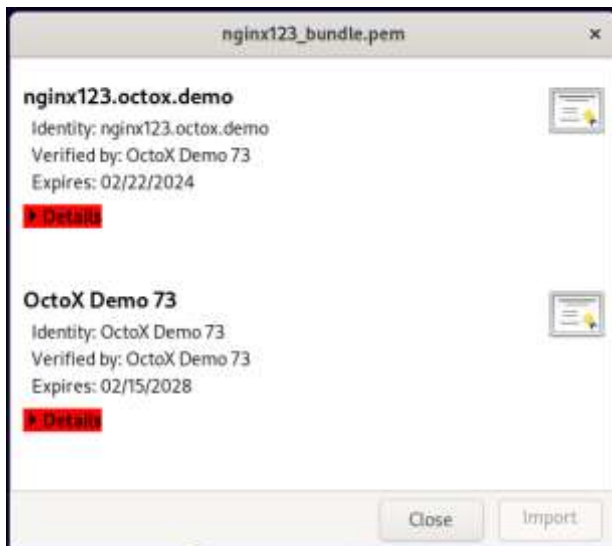
väljastatud veebiserveri sertifikaadi nginx123.pem ja selle väljastaja sertifikaadi O73.pem. Seda saame teha kas tekstiredaktoris sertifikaadid lihtsalt Base64 kodeeritud formaadis üksteise järele asetades või kasutades Ubuntu käsku „cat nginx123.pem OD73.pem > NGINX_123Bundle.pem“.



```
root@ubuntu2110: /home/uv/PKI
root@ubuntu2110:/home/uv/PKI# cat nginx2111.pem OctoX_D7.pem >nginx2111_Bundle.pem
root@ubuntu2110:/home/uv/PKI#
```

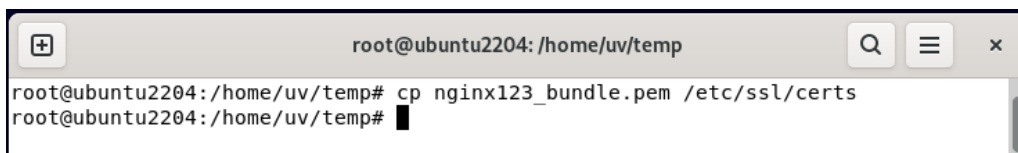
Pilt 14 - sertifikaatide koondamine ühte faili

Ubuntus avades näeb meie koondfail välja järgmine:



Pilt 15 - sertifikaadid on koondatud ühte faili

Paigaldamiseks sertifikaatide koondfaili korrektseesse konteinerisse /etc/ssl/certs käivitame terminalis käsu „cp nginx123_bundle.pem /etc/ssl/certs“.



```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204:/home/uv/temp# cp nginx123_bundle.pem /etc/ssl/certs
root@ubuntu2204:/home/uv/temp#
```

Pilt 16 - sertifikaatide koondfaili kopeerimine sertifikaatide konteinerisse

Lisaks peame korrektselt paigaldama ka väljastatud privaatvõtme. Väljastatud privaatvõti tuleb paigaldada kausta /etc/ssl/private. Selleks kasutame terminalis käsku „cp nginx123.key /etc/ssl/private“.



```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204:/home/uv/temp# cp nginx123.key /etc/ssl/private/
root@ubuntu2204:/home/uv/temp#
```

Pilt 17 - privaatvõtme paigaldamine

Nüüd on Nginx serveripoolsed sertifikaadi olemas ja korrektselt failisüsteemi paigaldatud.

Virtuaalse veebisaidi loomine

Loome enda demo-konfiguratsioonile eraldiseisva virtuaalse veebisaidi. Esmalt loome kausta /var/www/nginx123, kuhu paigaldame veebi sisu.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204:/home/uv/temp# mkdir /var/www
root@ubuntu2204:/home/uv/temp# mkdir /var/www/nginx123
root@ubuntu2204:/home/uv/temp#
```

Pilt 18 - kaustade loomine

Paigaldame loodud kausta mõne lihtsa ja äratuntava veebilehe nimega index.html.

Siis teeme valmis virtuaalse saidi konfiguratsioonifaili. Teeme uue faili nimega /etc/nginx/conf.d/nginx123.conf“ käsuga „nano /etc/nginx/conf.d/nginx123.conf“.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204:/home/uv/temp# nano /etc/nginx/conf.d/nginx123.conf
root@ubuntu2204:/home/uv/temp#
```

Pilt 19 – teeme uue konfiguratsioonifaili

Nüüd muudame uut konfiguratsioonifaili vastavalt oma soovidele. Lisame sinna järgmise sisu⁴:

```
# Start
server {
    listen 80;
    listen [::]:80;
    server_name nginx123.octox.demo;
    return 301 https://nginx123.octox.demo ;
}
server{
    # SSL configuration
    listen 443 ssl;
```

⁴ HTTP osa siin konfiguratsioonifailis ei ole tegelikult vajalik ja on toodud lihtsalt HTTP ->HTTPS ümbersuunamise näitena.



```
listen [::]:443 ssl;
root /var/www/nginx123;
index index.html;
server_name nginx123.octox.demo ;

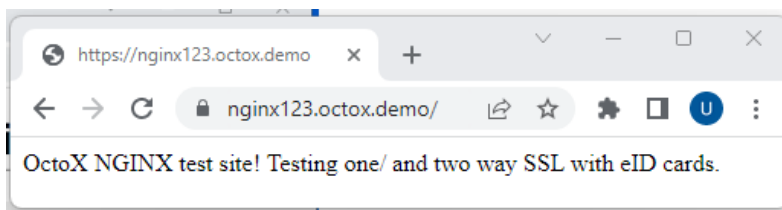
# Certificates
ssl_certificate /etc/ssl/certs/nginx123_bundle.pem;
ssl_certificate_key /etc/ssl/private/nginx123.key;

location / {
    try_files $uri $uri/ =404;
}
# End
```

Konfiguratsiooni korrektsust saame kontrollida käsuga „nginx -t“. Kui konfiguratsiooniga probleeme ei ole, siis aktiveerime uue konfiguratsiooni käivitades Nginx teenuse: „systemctl start nginx“. Juhul kui teenus juba töötab, saame selle taaskäivitada käsuga „systemctl reload nginx“.

Tulemus

Nüüd saame kasutada ühepoolset SSL-i saidi poole pöördumiseks (HTTPS ühendust). Samuti suunatakse meid ebatavaliselt aadressilt <http://nginx123.octox.demo> automaatselt aadressile <https://nginx123.octox.demo>.



Pilt 20 - Nginx veebiserver töötab ja kasutab ühepoolset SSL-i, kohandatud index.html!

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 nginx123.conf failile järgmised read SSL sektsiooni:

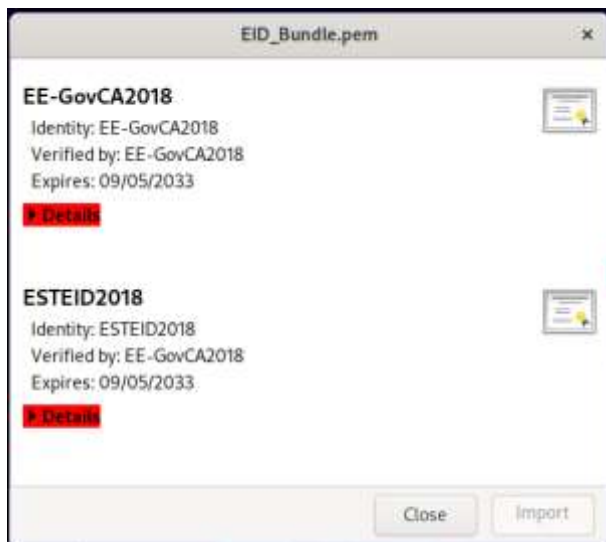
- ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;
- ssl_verify_client on;
- ssl_verify_depth 2;



```
# Certificates
ssl_certificate /etc/ssl/certs/nginx123_bundle.pem;
ssl_certificate_key /etc/ssl/private/nginx123.key;
ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;
ssl_verify_client on;
ssl_verify_depth 2;
```

Pilt 21 - selline on uus konfiguratsioonifaili Certificates/SSL osa

Nüüd loome uue tekstifaili EID_Bundle.pem⁵, kuhu lisame aktiivsed eID juur- ja kesktaseme sertifikaadid Base64 kodeeritud kujul (EE-GovCA2018, ESTEID2018). Selle faili abil filtreerime välja kõik sertimiskeskused, milliste alt väljastatud sertifikaate meie uus veebisait toetab. Samuti näidatakse kliendi pool vaid neid sertifikaate, mis on väljastatud eelloetletud ahelatest. Faili loomiseks saame kasutada cat käsku, aga töötab ka kopi-paste tekstiredaktorite vahel. Ubuntu failihalduris avatuna hakkab meie fail nägema välja järgmine:

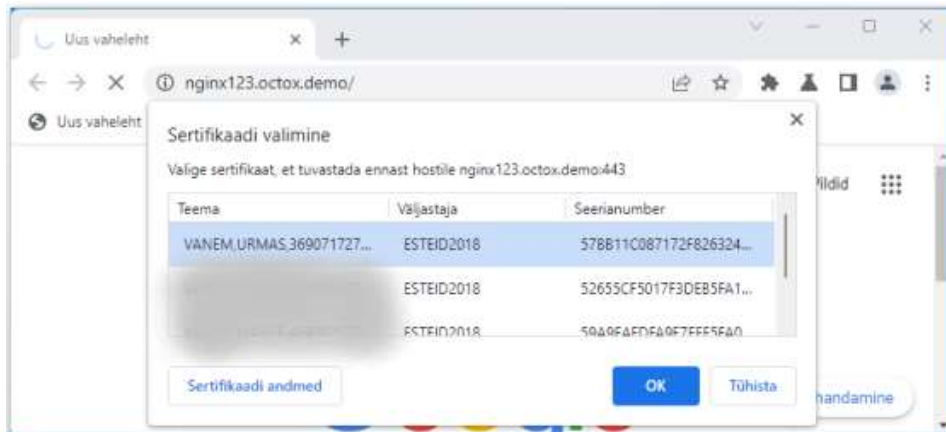


Pilt 22 – juur ja kesktaseme sertifikaadid ühes failis

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

Pöördudes peale muudatuse jõustumist uuesti veebisaidi nginx123.octox.demo poole, küsitakse meil kasutaja sertifikaati.

⁵ Allalaetav: https://installer.id.ee/media/id2019/EID_Bundle.pem



Pilt 23 - kasutaja sertifikaadi päring

Serveri soovitusel pakutakse meile välja kasutajasertifikaadid, milliste väljastajad on kirjeldatud failis EID_Bundle.pem. Peale sertifikaadi kinnitamist ja PIN-koodi sisestamist lastakse meid veebisaidile ligi. Kahepoolne SSL töötab!

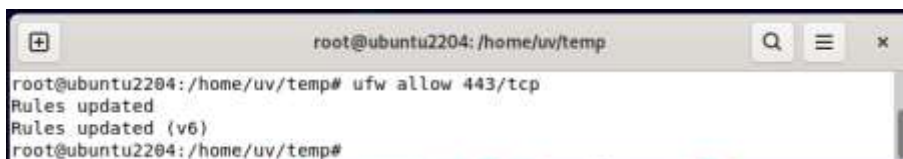
Ubuntu demokonfiguratsiooni fail selles dokumendis kirjeldatud muutujatega, k.a. osa lisakonfiguratsiooni peatüki all kirjeldatust, on alla laetav aadressilt https://installer.id.ee/media/id2019/NGINX_1.23.3_EID_Demo.conf.

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 mainida peatuda alloleval.

Tulemüüri reegli loomine, vajadusel

Tulemüüri reegli loomiseks tuleb terminalil käivitada käsk „ufw allow 'SOOVITAV REEGEL'“. Näiteks ainult https liikluse lubamiseks tuleb käivitada „ufw allow 443/tcp“.



Pilt 24 - https reegli loomine tulemüüris

Kui tulemüüri staatus on aktiivne (ufw enable), siis päring „ufw status“ näitab meile olemasolevaid reegleid.



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

To Action From
-- ---
443/tcp ALLOW Anywhere
443/tcp (v6) ALLOW Anywhere (v6)

root@ubuntu2204: /home/uv/temp#
```

Pilt 25 - tulemüür on aktiivne ja HTTPS liiklus veebiserveri poole on lubatud

OCSP põhise sertifikaadikontrolli kehtestamine⁶

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 ID Solutions AS*-i 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 NGINX veebisaidi SSL konfiguratsioonile lisama järgmised read:

```
ssl_ocsp leaf;
ssl_ocsp_cache off;
resolver 194.126.115.18;7
ssl_ocsp_responder http://ocsp.sk.ee;
```

```
# Certificates
ssl_certificate /etc/ssl/certs/nginx123_bundle.pem;
ssl_certificate_key /etc/ssl/private/nginx123.key;
ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;
ssl_verify_client on;
ssl_verify_depth 2;
ssl_ocsp leaf;
ssl_ocsp_cache off;
resolver 194.126.115.18;
ssl_ocsp_responder http://ocsp.sk.ee;
```

Pilt 26 – täiendatud SSL konfiguratsioon

Täiendatud konfiguratsiooni kehtestamiseks tuleb veebiteenus muidugi restartida. Seejärel kontrollitakse lisaks kasutajasertifikaadi ajalisele kehtivusele ja muudele omadustele ka selle usaldusväärsust vastu kesket OCSP teenust. Juhul, kui sertifikaadi kehtivus on tühistatud (ei ole

⁶ Sertifikaatide kehtivust on võimalik kontrollida ka sertifikaatide tühistusnimekirjade (CRL) abil, ent sellel me käesolevas dokumendis ei peatu, kuna peame OCSP-põhist lahendust paremaks.

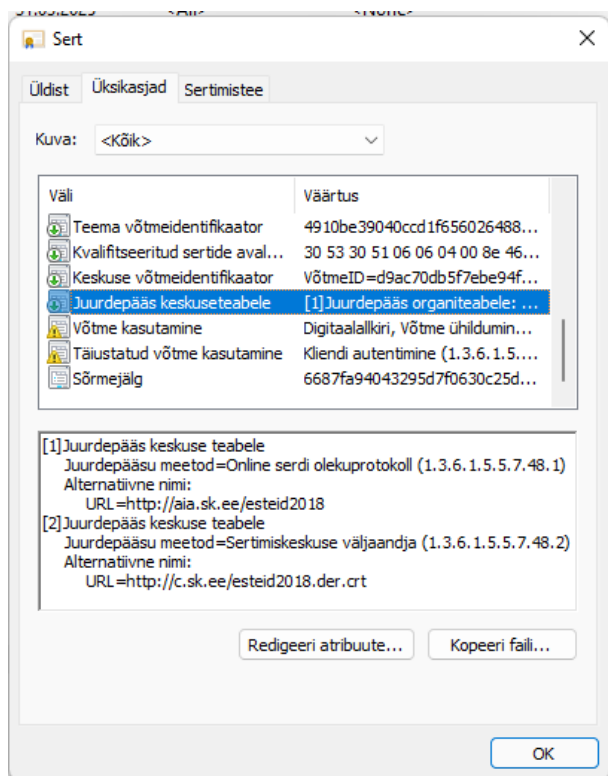
⁷ Juhul, kui `ssl_ocsp_responder` on konfiguratsioonis kirjeldatud, ei ole resolver määrangu olemasolu justkui nõutud – sertifikaadi kontroll töötab ka ilma selle määranguta. Seega võib selle määrangu kas samaks jätta, konfiguratsioonist eemaldada või asendada selle väärtus omapoolse nimeserveriga.



„GOOD“), ei ole sellega enam võimalik end vastu veebiteenust autentida. Märgin veelkord, et garanteeritud teenuse kasutamiseks peab olema sõlmitud eraldi leping SK ID Solutions AS-iga.

AIA OCSP teenus

Lisaks garanteeritud (tasulisele) OCSP lepingulisele teenusele pakub SK ID Solutions AS ka vaba ligipääsuga AIA OCSP teenust, mille puhul sertifikaate kontrollitakse pisut lihtsama ja tasuta OCSP teenuse vastu. AIA OCSP tee on ka SK sertifikaatidesse sisse kirjutatud: ESTEID2018 CA tasemelt väljastatud sertifikaadid: <http://aia.sk.ee/esteid2018>



Pilt 27 - ESTEID2018 AIA-OCSP tee sertifikaadis

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

```
ssl_ocsp leaf;  
ssl_ocsp_cache off;  
resolver 194.126.115.18;8
```

⁸ Resolver – asendage see soovi korral suvalise DNS serveriga, mis on võimeline avalikke DNS aadresse lahendada. Selliseks on tõenäoliselt ka teie enda sisevõrgu DNS server.



```
# Certificates
ssl_certificate /etc/ssl/certs/nginx123_bundle.pem;
ssl_certificate_key /etc/ssl/private/nginx123.key;
ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;
ssl_verify_client on;
ssl_verify_depth 2;
ssl_ocsp leaf;
resolver 194.16.115.18;
```

Pilt 28 - AIA OCSP konfiguratsioon on lisatud

Ülaltoodud konfiguratsiooni puhul loetakse OCSP serveri tee kasutaja sertifikaadist.

NGINX server soovituslikud turvasätted

SSL/TLS

Ubuntu platvormil töötav NGINX server versiooniga 1.23.3 võib toetada aegunud TLS versioone nagu TLS 1.0 või TLS 1.1. Tänapäeval soovitame tungivalt mitte kasutada TLS protokollid versioonist 1.2 madalamaid versioone. 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, et meie Nginx server toetaks vaid TLS protokollid versiooni 1.3, peame konfiguratsioonifaili lisama rea: “ssl_protocols TLSv1.3;”.

```
ssl_protocols TLSv1.3;
```

Pilt 29 – TLS 1.3 lubamine konfiguratsioonifailis

Toetamiseks ka TLS versiooni 1.2, tuleb konfiguratsioonireale lisada „TLSv1.2“.

Kui soovime sama muudatust kehtestada serveri tasemel, tuleb ssl_protocols käsku kohandada failis /etc/nginx/nginx.conf.

Rohkem infot TLS protokollid kasutamise soovitude kohta leiab RIA tellitud krüptograafiliste algoritmide elutsükli uuringust aadressil <https://www.id.ee/artikkel/krüptograafiliste-algoritmide-elutsükli-uuringud-2/>.

Šifrikomplektid (*cipher suites*)

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



TLS 1.2 puhul see päris nii ei ole. Nginx 1.23.3 versiooniga on vaikumisi kasutusel suur hulk erinevaid TLS šifreid⁹, milliseid näeme käsuga „openssl ciphers -v“.

Kui soovime ise täpsemalt määrata kasutatavaid šifrikomplekte, saame kasutada Nginx kaustapõhises konfiguratsioonifailis käsku `ssl_ciphers`. Siin omakorda saame kasutada kas eeldefineeritud aliasi või täpseid šifrikomplektide kirjeldusi.

Kindlat soovitus erinevate šifrikomplektide kasutamiseks ei ole veebisaidile esitatavaid tingimusi teadmata võimalik anda. Küll aga tuleb kindlasti eemaldada loendist ebatavalised šifrikomplektid. Mõistlik tundub kirjeldada konkreetseid lubatud šifrikomplektid TLS 1.2 kasutamiseks.

Näide:

- Kasutades järgmist käsurida konfiguratsioonifailis: `'ssl_ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384;'` – lubatakse vaid kirjeldatud šifrikomplektide kasutamine.

Alternatiivina saame kasutatavaid šifreid configureerida serveripõhiselt failis `/etc/nginx/nginx.conf` muutes selles parameetrit `ssl_ciphers`.

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

ssl_prefer_server_ciphers

Eelistamiseks serveri šifrikomplektide valikut kliendi omale, tuleb Nginx konfiguratsioonifailis defineerida määrang `ssl_prefer_server_ciphers` ja panna selle väärtuseks „on“!

Kasutajasertifikaatide lisafiltreerimine

Oluline! Kindlustamiseks, et meie veebiteenuse poole saavad pöörduda vaid „õiged“ kasutajad korrektsete sertifikaatidega, on tungivalt soovituslik kehtestada kasutajasertifikaatidele ka järgnev vastavuse kontroll:

- 1) sertifikaadis korrektse OID väärtuse olemasolu;
- 2) sertifikaadi väljastajaks peab olema ESTEID2018.

Paraku ei ole täna teada, kuidas esimest punkti NGINX puhul serveri tasemel saavutada on võimalik. Seega soovitame seda teha veebirakenduse tasemel.

Teise nõude täitmiseks saame luua konfiguratsiooni, kus ühendus katkestatakse, kui sertifikaat ei ole väljastatud meie poolt lubatud CA-de poolt. Selleks lisame konfiguratsioonifaili (serveri sektsiooni, näiteks SSL kirjeldusele järgnevalt) järgmised tingimused:

```
#Determine IMCA and cancel, if not trusted
    set $ocspr "";
    if ($ssl_client_i_dn = "CN=ESTEID2018,organizationIdentifier=NTREE-10747013,O=SK ID
    Solutions AS,C=EE") {
```

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.



```
        set $ocspr "http://aia.sk.ee/esteid2018";
    }
    if ($ocspr = "") {
        return 403;
    }
}
```

Peale ülaltoodud tingimuste lisamist sessioon katkestatakse juhul, kui kliendisertifikaat ei ole väljastatud ülalkirjeldatud ESTEID2018 CA poolt.

HTTPS serveri konfiguratsioon näeb nüüd välja järgmine (AIA OCSP teenusega):

```
# Start
server {
    listen 80;
    listen [::]:80;
    server_name nginx123.octox.demo;
    return 301 https://nginx123.octox.demo;
}
server{
    # SSL configuration
    listen 443 ssl;
    listen [::]:443 ssl;
    root /var/www/nginx123;
    index index.html;
    server_name nginx123.octox.demo;

    # Certificates
    ssl_certificate /etc/ssl/certs/nginx123_bundle.pem;
    ssl_certificate_key /etc/ssl/private/nginx123.key;
    ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;
    ssl_verify_client on;
    ssl_verify_depth 2;
    ssl_ocsp leaf;
    ssl_ocsp_cache off;
    resolver 194.126.115.18;
    ssl_protocols TLSv1.3;
    #ssl_ciphers ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers on;

    # Other recommended security and optimization settings
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always;
    ssl_session_cache shared:SSL:10m;
    ssl_session_timeout 1h;
    ssl_session_tickets on;

    #Determine IMCA and cancel, if not trusted
    set $ocspr "";
    if ($ssl_client_i_dn = "CN=ESTEID2018,organizationIdentifier=NTREE-10747013,O=SK ID Solutions AS,C=EE") {
        set $ocspr "http://aia.sk.ee/esteid2018";
    }
    if ($ocspr = "") {
        return 403;
    }

    location / {
        try_files $uri $uri/ =404;
    }
}
}
```

Pilt 30 – HTTPS serveri konfiguratsioon

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“): <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“ samalt 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 korrektset OID'i. Kui ei sisalda, siis ärge autentige. Täna teadaolevad OID-id on SK publitseerinud peatükis „Only accept certificates with trusted issuance policy“ samuti samal aadressil: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>

HTTP Strict Transport Security (HSTS) lubamine

HSTS teenuse Nginx veebisaidile konfigureerimiseks lisa rida “add_header Strict-Transport-Security “max-age=31536000; includeSubDomains; preload” always;” konfiguratsioonifaili.

```
# Other recommended security and optimization settings
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always;
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 1h;
ssl_session_tickets on;
```

Pilt 31 - HSTS aktiveerimine

Muud võimalused

Lisaks TLS ja šifrikomplektide häälestusele soovitame Nginx konfiguratsiooni turvalisusele pöörata tähelepanu ka järgmiste punktide vaates:

- Hoida operatsioonisüsteem uuendatuna.
- Hoida Nginx uuendatuna.
- Keelata serveri info presenteerimine.
- Keelata HTTP päringud.
- Installeerida ja konfigureerida Naxsi.
- Monitoorida Monit abil.
- Konfigureerida X-XSS kaitse.
- Konfigureerida X-Frame-Options.
- Konfigureerida X-Content-Type-Options.
- Konfigureerida Content Security Policy (CSP).
- ...

Ubuntu/Nginx SSL häälestus



Lihtne konfiguratsioonijuhend Eesti eID kaartide vaates

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