



Configuring two-way SSL using Estonian ID-cards in the Ubuntu Nginx web server

Document information	
Date of creation	08/02/2019
Client	Information System Authority
Author	Urmas Vanem, OctoX
Version	25.10/1

Version information		
Date	Version	Changes/Notes
08/02/2019	19.02/1	Public version.
28/02/2019	19.02/2	Added notes about the user certificate validity check. Default webpage removal. Changed by Urmas Vanem.
12/12/2019	19.12/1	Added recommendations for securing Nginx. Changed by Urmas Vanem.
30/12/2020	20.12/1	Ubuntu version is updated to 20.04.1. Nginx version is updated to 1.19.6. Configuration management is changed (sites... -> conf.d). Added the possibility of using OCSP-based revocation lists, recommended security settings, and the description for filtering out certificates issued by the wrong CAs. Changed by Urmas Vanem
13/01/2021	21.01/1	Added the demonstrative configuration file. Added HSTS configuration. Changed by Urmas Vanem
25/01/2021	21.01/2	Changed the HSTS recommendations. Changed the SSL/TLS and cipher recommendations. Added some additional security recommendations. Changed by Urmas Vanem
28/04/2021	21.04/1	Support for outdated ESTEID-SK 2011 certificates removed. Changed by Urmas Vanem
25/11/2021	21.11/1	Ubuntu version updated to Ubuntu Server 21.10. Nginx version updated to 1.21.4. Added guidance for ECC certificates. Updated TLS and cipher recommendations.

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards

		Changed by Urmas Vanem
22/02/2023	23.02/1	Ubuntu version is updated to Ubuntu Server 22.04 and the Nginx version is now 1.23.3. The virtual host configuration has also been updated. Changed by Urmas Vanem
18/12/2023	23.12/1	Removed the ESTEID-SK 2015 chain. Changed by Urmas Vanem
22/08/2024	24.08/1	Ubuntu version is updated to Ubuntu Server 24.04 and the Nginx version is now 1.27.1. Changed by Urmas Vanem
31.10.2025	25.10/1	Added Zetes chain, removed SK OCSP section Changed by Lauris Kaplinski



Intro

This guide describes:

- How to install and configure the Nginx 1.28.0 web server on Ubuntu Server 24.04.
- How to configure HTTPS (one-way SSL) in the web server.
- How to configure ID-card authentication (two-way SSL) in the web server using ID-cards.
- How to protect the web server.

In addition, we will look at other configuration options such as how to configure HTTP -> HTTPS redirection, etc.

Nginx installation and configuration

Installation

By default, Ubuntu 24.04 installs Nginx version 1.24. However, as we wish to use the latest version 1.28.0 in our guide, additional changes are required before installation.

To install Nginx 1.28.0 on Ubuntu 24.04 (in October 2024):

1. Run in terminal

```
add-apt-repository ppa:ondrej/nginx
```

A terminal window screenshot showing the command `sudo add-apt-repository ppa:ondrej/nginx` and its output. The output includes the PPA name, URI, suites, components, and a description of the package.

```
parallels@ubuntu-linux-24-04-desktop: ~  
parallels@ubuntu-linux-24-04-desktop:~$ sudo add-apt-repository ppa:ondrej/nginx  
PPA publishes dbgsym, you may need to include 'main/debug' component  
Repository: 'Types: deb  
URIs: https://ppa.launchpadcontent.net/ondrej/nginx/ubuntu/  
Suites: noble  
Components: main  
'  
Description:  
This branch follows latest NGINX Stable packages compiled against latest OpenSSL for  
HTTP/2 and TLS 1.3 support.  
BUGS&FEATURES: This PPA now has a issue tracker: https://deb.sury.org/#bug-reporting
```

Picture 1 – adding Nginx mainline package

2. Run in terminal:

```
apt update
```



```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# apt update
Hit:1 http://ee.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ee.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ee.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/apache2/ubuntu noble InRelease
Hit:6 https://ppa.launchpadcontent.net/ondrej/nginx-mainline/ubuntu noble InRelease
```

Picture 2 - renewing packages

3. Run in terminal:

```
apt install nginx-full
```

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# apt install nginx-full
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
  libnginx-mod-http-geoip2 libnginx-mod-http-subst-filter
  libnginx-mod-http-upstream-fair libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common
```

Picture 3 - installing Nginx

We can check Nginx version with command

```
nginx -v
```

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# nginx -v
nginx version: nginx/1.27.1
```

Picture 4 - as expected, Nginx version is 1.27.1

Configuration

Enabling one-way SSL

Creating the private key and the Certificate Signing Request (CSR)

Elliptic Curve Cryptography (ECC)

First, generate an ECC private key:

```
openssl ecparam -name secp384r1 -genkey -noout -out Nginx2404.key
```

Then, generate an ECC CSR:

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards

```
openssl req -new -key Nginx2404.key -out Nginx2404.csr -subj  
/C=EE/O=OctoX/CN=Nginx2404.octox.demo -reqexts SAN -config <(cat  
/etc/ssl/openssl.cnf <(printf  
"[SAN]\nsubjectAltName=DNS:Nginx2404.octox.demo,DNS:MYWEBSERVER.octo  
x.demo"))1
```

```
root@ubuntu-2404: /home/uv  
root@ubuntu-2404: /home/uv# openssl eparam -name secp384r1 -genkey -noout -out N  
ginx2404.key  
root@ubuntu-2404: /home/uv# openssl req -new -key Nginx2404.key -out Nginx2404.cs  
r -subj /C=EE/O=OctoX/CN=Nginx2404.octox.demo -reqexts SAN -config <(cat /etc/ss  
l/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS:Nginx2404.octox.demo,DNS:MYWEB  
SERVER.octox.demo"))  
root@ubuntu-2404: /home/uv#
```

Picture 5 – creating private key and certificate request file

Arguments on the yellow background:

1. Nginx2404.key is the private key of the certificate;
2. Nginx2404.csr is the CSR for the certificate authority (CA);
3. CN=Nginx2404.octox.demo is the common name for the certificate;
4. DNS:Nginx2404.octox.demo and DNS:MYWEBSERVER.octox.demo are SAN DNS names for the certificate. These names must correspond to the actual address of the website². The names must also be resolvable in name services.

The contents of the CSR can be viewed by running

```
openssl req -in Nginx2404.csr -noout -text
```

¹ In addition to the certificate attributes C, O, and CN described on the command line, it is also possible to describe the attributes L, OU, and S if desired. However, only CN can also be used.

² Modern browsers do not trust websites where at least one SAN DNS name is not equal to the actual address of the website.



```
root@ubuntu-2404: /home/uv
root@ubuntu-2404:/home/uv# openssl req -in Nginx2404.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: C = EE, O = OctoX, CN = Nginx2404.octox.demo
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (384 bit)
    pub:
      04:83:8a:77:21:33:00:ac:6a:66:28:f4:7e:8e:98:
      fa:52:09:ed:bb:83:f9:98:ee:24:3b:48:b1:e2:ad:
      ae:1d:57:78:b1:9a:5c:c7:9c:4c:cb:95:f9:ff:b1:
      89:4f:d8:c9:e1:39:0e:5d:ac:c6:d3:92:64:39:23:
      5c:d0:fc:0e:38:17:22:3c:bb:e0:fb:ca:2c:8e:55:
      65:2b:7c:56:6a:55:4a:b8:ae:a4:8c:e5:81:b7:a9:
      d6:e4:5a:1a:aa:af:37
    ASN1 OID: secp384r1
    NIST CURVE: P-384
  Attributes:
    Requested Extensions:
      X509v3 Subject Alternative Name:
        DNS:Nginx2404.octox.demo, DNS:MYWEBSERVER.octox.demo
  Signature Algorithm: ecdsa-with-SHA256
  Signature Value:
```

Picture 6 – the contents of the CSR file

RSA

The following section on creating RSA CSR is left here from the previous version of the document if somebody wants to continue using certificates based on the RSA algorithm. In the following chapters of the document, certificates based on the ECC algorithm are described.

Create a CSR and a private key with the command

```
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/openss
l.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'
-----
```

Picture 7 – generating the private key and CSR

Parameters on yellow background:

1. Nginx20PRIV.key is the private key of the certificate;
2. Nginx20.csr is the CSR for the CA;



3. Nginx20.kaheksa.xi is the subject name for the certificate;
4. Nginx20.kaheksa.xi and Nginx22.kaheksa.xi are certificate SAN DNS names. These names must correspond to the actual address of the website³. The names must also be resolvable in name services.

The contents of the CSR can be viewed with the command

```
openssl req -in NGINX20.csr -noout -text
```

```
root@Ubuntu2028:/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:
      47:78:aa:3e:c8:4c:50:62:5b:6c:2a:49:ea:51:81:
      60:5c:94:2c:06:1d:78:70:eb:41:88:6c:09:c8:2f:
      e4:d5:bb:2f:fb:0c:2f:9d:0c:42:66:b5:de:91:e3:
      60:62:ff:94:11:21:aa:de:bb:52:b0:20:a6:ff:b4:
      c3:92:0a:5b:b5:fc:2f:08: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:90:06:92:2d:a6:ea:d7:d2:09:8f:51:e8:b0:52:
      07:b1:1e:5e:ca:65:f3:d4:09:52: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:
```

Picture 8 – the contents of the CSR

Ordering and installing an SSL certificate

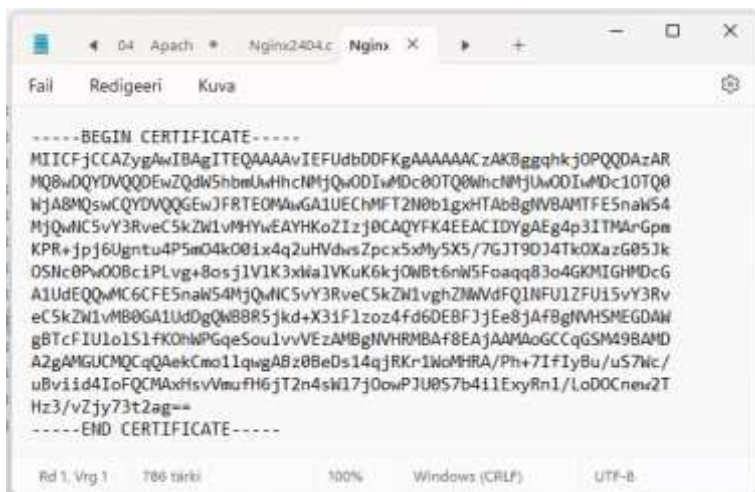
The CSR Nginx2404.csr should be sent to trustworthy CA. In the demo environment, the certificate issuer is the test CA. Signed certificate is issued in Base64 encoded format.

³ Browsers do not trust websites where at least one SAN DNS name is not equal to the actual address of the website.

Ubuntu/Nginx SSL configuration

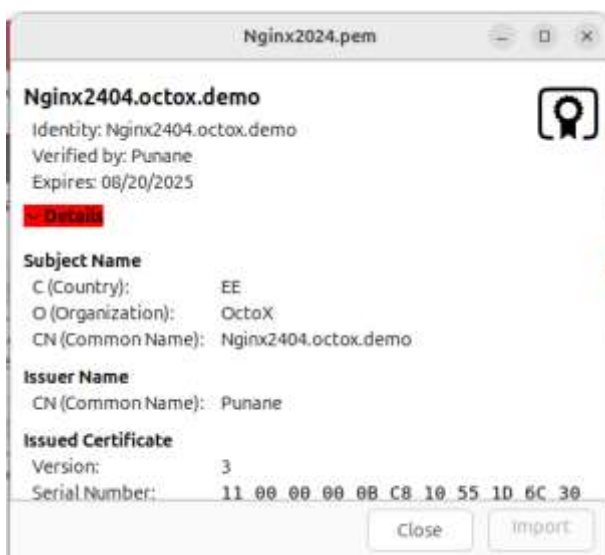


Simple configuration guide for Estonian ID-cards



Picture 9 – Base64 format certificate in a text redactor

In Ubuntu, the certificate looks like this:



Picture 10 – ECC certificate in Ubuntu

The certificate also includes the algorithm and alternative SAN DNS names of the subject:

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards



Picture 11 – the algorithm and SAN DNS names

As you can see, certificate issuer is a CA named ‘Punane’. Now, you need to obtain the issuer CA’s certificate in Base64 encoded format and save it to the user’s home folder in Ubuntu as Punane.pem.

Consolidate all certificates used for one-way SSL into one file; thereat, the first certificate in the file has to be the web server certificate. In this example, consolidate the issued web server certificate Nginx2404.pem and its CA certificate, Punane.pem. This can be done in a text redactor (by placing the certificates in Base64 encoded format after one another) or with the command

```
cat Nginx2404.pem Punane.pem > Nginx2404_Bundle.pem
```



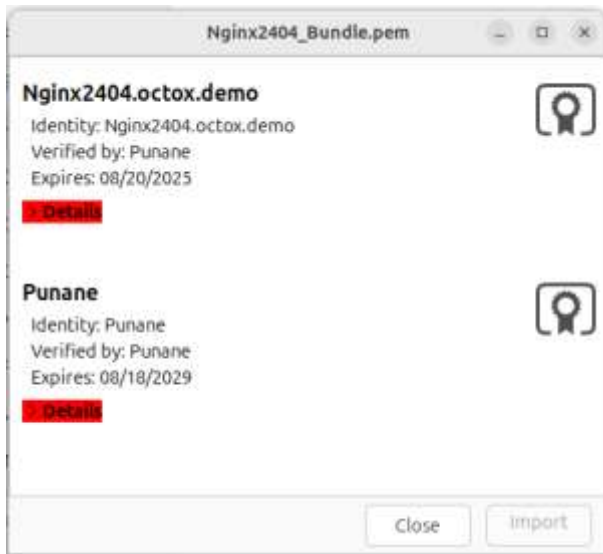
Picture 12 – consolidating certificates into one file

When opened in Ubuntu, the file looks like this:

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards



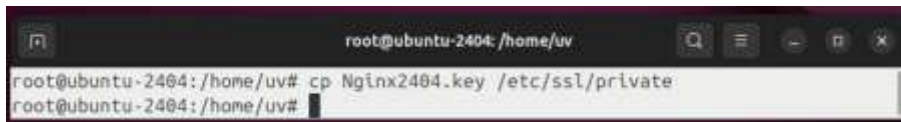
Picture 13 – the certificates are consolidated into a single file

Place the bundled certificate file **Nginx2404_Bundle.pem** to the **/etc/ssl/certs** folder.



Picture 14 – copying the bundled file to the certificate's container

In addition, you need to move the certificate private key to the **/etc/ssl/private** folder.

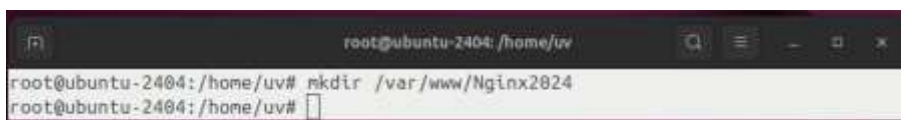


Picture 15 – copying the private key

Now, all the certificates and the private key needed by Nginx for one-way SSL have been correctly installed.

Creating a virtual website

Create a separate virtual website for your configuration. First, create a home folder named **/var/www/Nginx2404** for the content of the website.

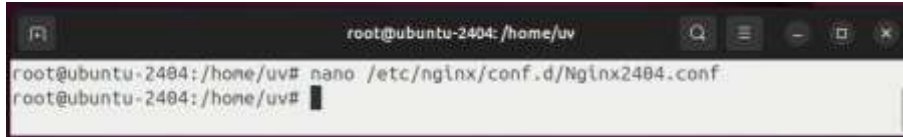


Picture 16 – creating folder for the website

Put a simple and recognizable webpage named **index.html** into the folder you created.



Then, prepare the configuration file for the new virtual website. Create a new file named **/etc/nginx/conf.d/Nginx2404.conf** (e.g. with the command **nano /etc/nginx/conf.d/Nginx2404.conf**).



Picture 17 – creating a new configuration file

Now, change the new configuration file as you wish. Paste the following configuration in it⁴:

```
# Start
server {
    listen 80;
    listen [::]:80;
    server_name Nginx2404.octox.demo;
    return 301 https://Nginx2404.octox.demo;
}
server{
    # SSL configuration
    listen 443 ssl;
    listen [::]:443 ssl;
        root /var/www/Nginx2404;
        index index.html;
        server_name Nginx2404.octox.demo;

    # Certificates
    ssl_certificate /etc/ssl/certs/Nginx2404_Bundle.pem;
    ssl_certificate_key /etc/ssl/private/Nginx2404.key;

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

You can check the correctness of the configuration by running the command **nginx -t**. If there are no problems with the configuration, you can activate the web service by starting nginx:

```
systemctl start nginx
```

If nginx is already running, you can apply the changes by restarting nginx with the terminal command

```
systemctl reload nginx
```

⁴ The HTTP sections in the configuration file are not necessary and are shown here as an example of the HTTP -> HTTPS redirection.



Result

Now, the new website can be accessed by one-way SSL. In addition, all HTTP requests to the site <http://Nginx2404.octox.demo> are automatically redirected to the HTTPS site <https://Nginx2404.octox.demo>.



Picture 18 – the Nginx web server is working and using one-way SSL

Requiring two-way SSL

If you wish to allow website access by authenticating with an Estonian ID-card, you need to supplement the existing configuration slightly.

Add the following lines to the SSL section of the file Nginx2404.conf:

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

```
# Certificates  
ssl_certificate /etc/ssl/certs/Nginx2404_Bundle.pem;  
ssl_certificate_key /etc/ssl/private/Nginx2404.key;  
ssl_client_certificate /etc/ssl/certs/EID_Bundle.pem;  
ssl_verify_client on;  
ssl_verify_depth 2;
```

Picture 19 – updated configuration file, SSL section

Now, create a new text file named **EID_Bundle.pem**⁵, which includes the eID root and intermediate certificates (EE-GovCA2018, ESTEID2018, EEGovCA2025, ESTEID2025) in Base64 encoded format. With this file, you can filter out all CA's whose certificates are supported by the new website. The user will only see the certificates from those chains. The 'cat' command can be used to create this file, but it also works as copy-and-paste between text redactors. When opened in Ubuntu, the file looks like this:

⁵ Available: EID_Bundle.pem

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards



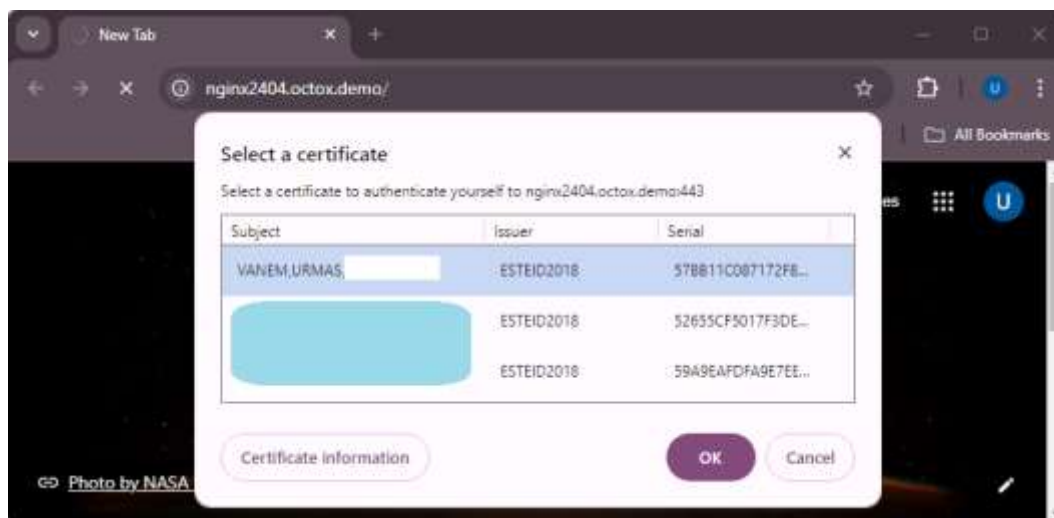
Picture 20 – root and intermediate certificates in one file

Save the file as EID_Bundle.pem and copy it to the folder `/etc/ssl/certs`. Restart the Nginx web server with the command

```
systemctl reload nginx
```

to activate the change in the web server.

After accessing the website `nginx2404.octox.demo` now, a user certificate is required:



Picture 21 – client certification request



The server suggests certificates to the user, the issuers of which are described in the file EID_Bundle.pem. After confirming the certificate and entering the PIN, the user can access the website – two-way SSL works.

The demonstrative Nginx configuration file with the settings in this document, including some additional configuration options, is downloadable from https://installer.id.ee/media/id2019/Nginx_1.27.1_EID_Demo.conf.

Additional configuration options

The purpose of this document is not to give exact guidance on how to optimize or protect websites, but to show how to configure two-way SSL for Estonian ID-cards. However, you should take into account the following.

Firewall rules (if necessary)

For creating a firewall rule, run the command

```
ufw allow 'DESIRABLE RULE'
```

on the terminal. For example, to allow HTTPS traffic only, run

```
ufw allow 443/tcp
```

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404:/home/uv# ufw enable
Firewall is active and enabled on system startup
root@ubuntu-2404:/home/uv# ufw allow 443/tcp
Rule added
Rule added (v6)
root@ubuntu-2404:/home/uv#
```

Picture 22 – activating firewall and creating a https rule

If the firewall is active (**ufw enable**), running the command **ufw status** in the terminal shows the active rules.

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404:/home/uv# ufw status
Status: active

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

root@ubuntu-2404:/home/uv#
```

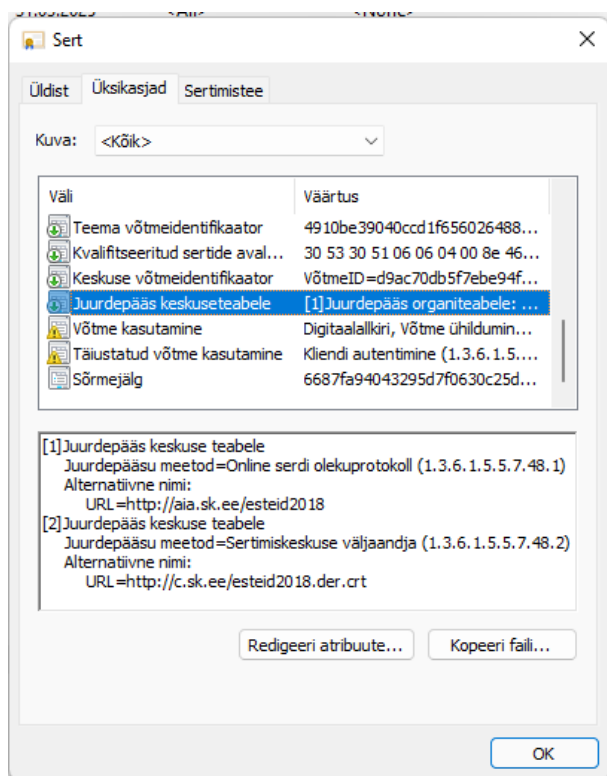
Picture 23 – the firewall is active and only HTTPS traffic is enabled



Checking the status of the user's certificate against the OCSP service⁶

Using the OCSP (Online Certificate Status Protocol) service, you can check the revocation status of client certificates practically in real time. With every client authentication attempt, the web server sends a query to the OCSP service, which responds with the certificate status.

SK and Zetes offer a free-access (free-of-charge) AIA OCSP service. For certificates issued under the ESTEID2018 and ESTEID2025 CA, AIA OCSP service location is already included in the certificate (<http://aia.sk.ee/esteid2018>, <http://ocsp.eidpki.ee>).



Picture 24 – ESTEID2018 AIA OCSP address in the certificate

To enable the user certificate validity check against the AIA OCSP service, you need to add the following lines to the SSL configuration of Nginx:

```
ssl_ocsp leaf;  
ssl_ocsp_cache off;  
resolver 194.126.115.18; 7
```

⁶ The certificate check is also doable with certificate revocation lists (CRL), but this is not covered in this document, as the OCSP-based solution is preferred.

⁷ Resolver – replace the IP address here with any DNS server that can resolve public DNS addresses. You can also use the DNS server of your intranet for this.



```
# 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;
```

Picture 25 – the AIA OCSP configuration is added

With the configuration shown above, the address of the OCSP service is taken from the client certificate.

Recommended security settings for Nginx

SSL/TLS

Nginx version 1.24 running on Ubuntu can support old TLS versions like TLS 1.0 or TLS 1.1 by default. It is no longer recommended to use TLS protocols with a version number lower than TLS 1.2. TLS version 1.3 has also been in use for a while.

If there is no specific requirement to allow TLS 1.2, it is recommended to only use TLS 1.3. While TLS 1.2 is very stable and secure with the correct configuration, TLS version 1.3 is faster, more secure by default, and needs less configuration. In standard situations, TLS 1.2 should be enabled only if really needed, and if it is enabled, it is mandatory to allow only secure cipher suites and extensions.

To configure Apache to support only TLS protocol version 1.3, you need to add the following line to the Nginx configuration file:

```
ssl_protocols TLSv1.3;
```

```
ssl_protocols TLSv1.3;
```

Picture 26 – enabling only TLS version 1.3 in the configuration file

To support TLS versions 1.2 and 1.3, add **TLSv1.2** to configuration line.

If you want to make the change at the server level, modify the parameter `ssl_protocols` in the file **/etc/nginx/nginx.conf**.

More information about the recommendations for the use of the TLS protocol can be found in the cryptographic algorithms life cycle report ordered by RIA at <https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/>.

Cipher suites

All TLS 1.3 cipher suites are currently considered safe, no additional configuration is required for security considerations for this protocol.



TLS 1.2 is different. There are many different TLS cipher suites available with Nginx version 1.24⁸, which can be viewed with the command

```
openssl ciphers -v
```

If you wish to configure the available cipher suites used with TLS 1.2 in more detail, you can use the command `ssl_ciphers` in the Nginx configuration file. Here, you can use predefined aliases or exact cipher suite descriptions.

It is impossible to give an exact recommendation for configuring cipher suites without knowing the requirements applicable to the webpage. However, non-secure cipher suites must be removed from the list. It is reasonable to describe the specific enabled cipher suites for TLS 1.2.

Example:

- Using following command line in the configuration file:
`ssl_ciphers ECDHE-ECDSA-AES256-GCM-SHA384 : ECDHE-RSA-AES256-GCM-SHA384;`
only the described cipher suites are allowed.

You can also configure cipher suites at the server level by modifying the parameter `ssl_ciphers` in the file `/etc/nginx/nginx.conf`.

More information about the recommendations for the use of the cipher suites can be found in the cryptographic algorithms life cycle report ordered by RIA at <https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/>.

ssl_prefer_server_ciphers

Preferring server ciphers over user ciphers can be enabled with `ssl_prefer_server_ciphers`. Set its value to 'on' in the configuration file.

Additional filtering of client certificates

Important! To guarantee access to our web service only for users with the right certificates, it is strongly recommended to add the following additional requirements in the server configuration:

- 1) The certificate must include the correct OID value;
- 2) The certificate issuer must be ESTEID2018 or ESTEID2025.

We do not currently know how to check the existence of the correct OID at the server level with Nginx. So, it is recommended to do it at the web application level.

⁸ The ciphers of other protocols are not covered in this chapter, because protocols older than version 1.2 should be disabled and version 1.3. is currently preferred.



To implement the second recommendation above, you can check the end-user certificate issuer and reject the connection if it is not issued by a CA enabled in the server. To implement it, add the following conditions to the configuration file (server section, e.g. after the SSL description):

```
#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 $ocspr "http://aia.sk.ee/esteid2018";
    }
    if ($ssl_client_i_dn = "CN=ESTEID2025, organizationIdentifier=NTREE-17066049, O=Zetes Estonia OÜ, C=EE") {
        set $ocspr "http://ocsp.eidpki.ee";
    }
    if ($ocspr = "") {
        return 403;
    }
```

After adding the conditions above to our configuration, any session will be cancelled if the user certificate is not issued from the CA described above – ESTEID2018 or ESTEID2025.

Notes:

- If you are using another feature to filter network traffic, the secure configuration should be implemented there, too. SK has published information about the F5 configuration in the chapter 'Only accept certificates with trusted key usage' in the following article: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>
- SK's recommendations for secure ID-card authentication are published here in the chapter 'Defense: implement ID-card authentication securely': <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>
- The recommended method for avoiding incorrect certificates is using OIDs in the certificates. Unfortunately, there is currently no method for doing this at the server level. If possible, open the certificate at the web application level, check for a correct OID in the certificate, and if there is none, reject the authentication request. All currently known OIDs are listed in the chapter 'Only accept certificates with trusted issuance policy' in the following article published by SK: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>

Enabling HTTP Strict Transport Security (HSTS)

To enable HSTS for the Nginx webpage, add the line

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload" always;
```

to the configuration file.



```
# Other recommended security and optimization settings.
ssl_prefer_server_ciphers on;
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;
```

Picture 27 – activating HSTS

Additional possibilities

In addition to TLS and cipher suite configuration, you should pay attention to the security of the Nginx server from the following aspects:

- Keep the operating system up to date.
- Keep Nginx up to date.
- Disable presenting server information.
- Disable HTTP requests.
- Install and configure Naxsi.
- Monitor with Monit.
- Configure X-XSS Protection.
- Configure X-Frame-Options.
- Configure X-Content-Type-Options.
- Configure Content Security Policy (CSP).
- ...

The above is a sample list of ways to improve Nginx security. Detailed recommendations are available online: <https://www.google.com/search?q=how+to+secure+nginx+server>.



Appendix

EID_Bundle.pem

```
EE-GovCA2018
-----BEGIN CERTIFICATE-----
MIIE+DCCBFmgAwIBAgIQMLOWlXoR0oFbj52nmRsneZAKBggqhkJOPQQDBDBaMQsw
CQYDVQQGEwJFRTEbMBkGA1UECgwSU0sgSUQgU29sdXRpb25zIEFTMRcwFQYDVQRh
DA5OVFJFRS0xMDC0NzAxMzEVMBMGA1UEAwwMRUUtR292Q0EyMDE4MB4XDTE4MDkw
NTA5MTEwM1oXDTMzMDkwNTA5MTEwM1owWjELMAkGA1UEBhMCRUUXGzAZBgNVBAoM
E1NLEIEIEIFNvbHV0aW9ucyBBUzEXMBUGA1UEYQwOT1RSRUUtMTA3NDcwMTMxFTAT
BgNVBAMMDEVFLUdvdnkNBMjAxODCBmzAQBgqhkJOPQIBBgUrgQQAiwoBhgAEAMcb
/dmAcVo/b2azEPS6CFW7fEA2KuHKC53D7ShVNvLz4QUjCdTXjds/4u99jUoYEQec
luVVzMlgEJR1nkN2eOrLAZYxPjwG5Hi1liZEyW9QKVdeEgyvzhWWTNHGjV3HdZRv
7L9o4533PtJAYqJ90tIB9zCCAfmwCAYGBACPEGECMAkGBwQAI+xAAQIwMgYLKwYB
BAGDkSEBAQEwIzAhBggrBgEFBQCARYVaHR0cHM6Ly93d3cuc2suZWUvQ1BTMA0G
CysGAQQBg5EhAQECMA0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5EhAQEFMA0GCysG
AQQBg5EhAQEGMA0GCysGAQQBg5EhAQEHMA0GCysGAQQBg5EhAQEDMA0GCysGAQQB
g5EhAQEEA0GCysGAQQBg5EhAQEIMA0GCysGAQQBg5EhAQEJMA0GCysGAQQBg5Eh
AQEKMA0GCysGAQQBg5EhAQELMA0GCysGAQQBg5EhAQEMMA0GCysGAQQBg5EhAQEN
MA0GCysGAQQBg5EhAQEOMA0GCysGAQQBg5EhAQEPMA0GCysGAQQBg5EhAQEQMA0G
CysGAQQBg5EhAQERMA0GCysGAQQBg5EhAQESMA0GCysGAQQBg5EhAQETMA0GCysG
AQQBg5EhAQEU0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5F/AQEDMA0GCysGAQQBg
5F/AQEEA0GCysGAQQBg5F/AQEFMA0GCysGAQQBg5F/AQEGMA0GCysGAQQBg5Eh
CgEwIzAhBggrBgEFBQCARYVaHR0cHM6Ly93d3cuc2suZWUvQ1BTMBGCCSQAQUF
BwEDBAwwCjAIBgYEAISGAQEwCgYIKoZIzj0EAwQDgYwAMIGIAkIBk698EquetY9Tt
6Hw050CfzdIjKmlfCI34xkdU7J+wz1tNVu2tHJwEhdsH0e92i9699sRDp1RNP1Vh
4XFJzI3oQFQCQgVxmcuVnsy7NUScDZ0erwovmbFOsNxELCANxNSWx5xMqzEiHv8
46opxu10UFDIBBPzkbBenL4h+g/WU71G78fIhA==
-----END CERTIFICATE-----
ESTEID2018
-----BEGIN CERTIFICATE-----
MIIFVzCCBLigAwIBAgIQdUf6rBR0S4tbo2bU/mZV7TAKBggqhkJOPQQDBDBaMQsw
CQYDVQQGEwJFRTEbMBkGA1UECgwSU0sgSUQgU29sdXRpb25zIEFTMRcwFQYDVQRh
DA5OVFJFRS0xMDC0NzAxMzEVMBMGA1UEAwwMRUUtR292Q0EyMDE4MB4XDTE4MDkw
NTA5MTEwM1oXDTMzMDkwNTA5MTEwM1owWDELMAkGA1UEBhMCRUUXGzAZBgNVBAoM
E1NLEIEIEIFNvbHV0aW9ucyBBUzEXMBUGA1UEYQwOT1RSRUUtMTA3NDcwMTMxFTAT
BgNVBAMMCKVTVEVJRDIwMTgwgZswEAYHkoZiZj0CAQYFK4EEACMDgYYABAHHOB1v
7URPYP1yHhOb7RA/YBDbtgyNSVMqYdxnFrKHUXh6tFkghvHuA1k2DSom1he5kqh
B5VspDembwWDJBOQWGOI/0t3EtcCLyjem7F9xOPdzUbZaIbpNRHpQgVbPFX0xpL
TgW27MpImH08DHBWfPeAaNX3eUpD4gC5cvhsK0RFEqOCAx0wggMZMB8GA1UdIwQY
MBaAFH4pVuc0knhOd+FvLjMqmHHB/TSfMB0GA1UdDgQWBbTzrHDbx36+1Pig5L5H
otA0rZoaqEjAObgNVHQ8BAf8EBAMCAQYwEgYDVR0TAAQH/BAgwBgEB/wIBADCCAc0G
A1UdIASCAcQwggHAMAGBgQAj3oBAJAjBgcEAIvsQAECMDIGCysGAQQBg5EhAQEB
MCMwIQYIKwYBBQUHAgEWFwh0dHBzOi8vd3d3LnNrLmV1L0NQUzANBg9rBgEEAYOR
IQEBjANBg9rBgEEAYORfwbEBATANBg9rBgEEAYORIQEBBTANBg9rBgEEAYORIQEB
BjANBg9rBgEEAYORIQEBBzANBg9rBgEEAYORIQEBBzANBg9rBgEEAYORIQEBBDAN
Bg9rBgEEAYORIQEBBDANBg9rBgEEAYORIQEBCTANBg9rBgEEAYORIQEBcANBg9r
BgEEAYORIQEBcANBg9rBgEEAYORIQEBDDANBg9rBgEEAYORIQEBDTANBg9rBgEE
AYORIQEBDjANBg9rBgEEAYORIQEBDzANBg9rBgEEAYORIQEBEDANBg9rBgEEAYOR
IQEBETANBg9rBgEEAYORIQEBEjANBg9rBgEEAYORIQEBEzANBg9rBgEEAYORIQEB
FDANBg9rBgEEAYORfwbEBAjANBg9rBgEEAYORfwbEBAzANBg9rBgEEAYORfwbEBBDAN
Bg9rBgEEAYORfwbEBBTANBg9rBgEEAYORfwbEBBjAqBgNVH5SUBAf8EIDAeBggrBgEF
BQCDCQYIKwYBBQUHAgIGCCSQAQUFBwMEMGoGCCSQAQUFBwEBBF4wXDAPBggrBgEF
BQCwAYYdaHR0cDovL2FpYy55ay51ZS91ZS1nb3ZjYTYwMTgwgLwYIKwYBBQUHMAKG
I2h0dHA6Ly9jLnNrLmV1L0VFLUdvdnkNBMjAxOC5kZXIuYy93d3cuc2suZWUvQ1BTMA0G
CysGAQQBg5EhAQEFMA0GCysGAQQBg5EhAQEHMA0GCysGAQQBg5EhAQEDMA0GCysGAQQB
g5EhAQEEA0GCysGAQQBg5EhAQEIMA0GCysGAQQBg5EhAQEJMA0GCysGAQQBg5Eh
AQEKMA0GCysGAQQBg5EhAQELMA0GCysGAQQBg5EhAQEMMA0GCysGAQQBg5EhAQEN
MA0GCysGAQQBg5EhAQEOMA0GCysGAQQBg5EhAQEPMA0GCysGAQQBg5EhAQEQMA0G
CysGAQQBg5EhAQERMA0GCysGAQQBg5EhAQESMA0GCysGAQQBg5EhAQETMA0GCysG
AQQBg5EhAQEU0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5F/AQEDMA0GCysGAQQBg
5F/AQEEA0GCysGAQQBg5F/AQEFMA0GCysGAQQBg5F/AQEGMA0GCysGAQQBg5Eh
CgEwIzAhBggrBgEFBQCARYVaHR0cHM6Ly93d3cuc2suZWUvQ1BTMBGCCSQAQUF
BwEDBAwwCjAIBgYEAISGAQEwMAYDVR0fBCKwJzAlc0GIIYYfaHR0cDovL2Muc2suZWUv
RUUtR292Q0EyMDE4LmNybdAKBggqhkJOPQQDBA0BjAAwGyGcQgDeuUY4HczUbfKS
002HZ88gclgYdzthGqLLENyTmtXE6dMBRnCbGUmhBCAA0mJSHbyFJ8W9ikLiSyurm
kJM0hDE9KgJCASOqA405Ia5nKjTJPNsHQ1Mi7KZsIcTH0oBccx+54N8ZX1MgBozJ
mT59rZY/2/OeE163BAwD0UduQUAnMPP6+W3Vd
-----END CERTIFICATE-----
EEGovCA2025
```

Ubuntu/Nginx SSL configuration



Simple configuration guide for Estonian ID-cards

```
-----BEGIN CERTIFICATE-----
MIIC1jCCAhygAwIBAgIUkKbXJo8FWjthNs7Hgduq1RiXqswsCgYIKoZIZj0EAwMw
WDEUMBIGA1UEAwLRUVHb3ZDQTIwMjUxZjZAVBgNVBGEtMk5UUKVFLTE3MDY2MDQ5
MRowGAYDVQQKDBFhZXRlcjBFc3RvbmlhIE/DnDELMAkGA1UEBhMCRUUwHhcNMjUw
NTA2MDgxODEzWWhcNDAwNTA1MDgxODEzWjBYMRQwEgYDVQDDAtFRUdvdKNBMjAy
NTEwMjUxZjZAVBgNVBAoMEVpldGVzIEVzZG9u
aWEgT8OcmQswCQYDVQQGEwJFRTRBMBAGByqGSM49AgEGBSuBBAAiA2IABH0zMU4D
UN/Ay6gUdWzMUdAYFau0flpuuicO2bfK7kHNGw+psRRn6DaF/4cVQd8qHxbDF2x
N4jJf1bSpQHlsc2RZHSCi8qb4E9GmB5MDovVxiXnBHOOW3+55Qm/BfvcwaOBpjCB
ozASBgNVHRMBAf8ECDAGAQH/AgEBMB8GA1UdIwQYMBaAFKqAqJsPu0umfsUC9HLN
LPG1Kdm3MD0GA1UdIAQ2MDQwMgYEVRO0gADAqMCgGCCsGAQUFBwIBFhxodHRwczov
L3JlcG9zaXRvcnkuzWlkcGtpLmVlLMB0GA1UdDgQWBBSqgKibD7tLpn7FAvRyzSzx
pSnZtzA0BgNVHQ8BAf8EBAMCAQYwCgYIKoZIZj0EAwMDAAAwZQIwOy8+eV+yYNxt
XcEEedOuQd60071XucK3W4cDewxEoEXb4iTYFswWUZq3DacfmE+/AjEAkzHeNdrU
QqKfvqTFB3eNRmMycNcnJ3rmGe37u9zgh8wnQUuMhUCLOGxeRcK4NV9I
-----END CERTIFICATE-----
ESTEID2025
-----BEGIN CERTIFICATE-----
MIIDDzCCApagAwIBAgIUUFQrcGtK7/jCP+GyAOTFvbg1G1cwCgYIKoZIZj0EAwMw
WDEUMBIGA1UEAwLRUVHb3ZDQTIwMjUxZjZAVBgNVBGEtMk5UUKVFLTE3MDY2MDQ5
MRowGAYDVQQKDBFhZXRlcjBFc3RvbmlhIE/DnDELMAkGA1UEBhMCRUUwHhcNMjUw
NTA3MTMyMDA3WWhcNDAwNTA2MDgxODEzWjBYMRMwEQYDVQDDApFU1RFSUQyMDI1
MRcWFQYDVQRhDA5OVFFJFRS0xNzA2NjA0OTEaMBgGA1UECgwRWmV0ZXNMgRXN0b25p
YSBFPw5wCzAJBgNVBAYTAkVFMHYwEAYHKOZIZj0CAQYFK4EEACIDYgAEdSEmb1An
xN7G22CCEQ3ts2YZNiETUZP4Vc4iObhmL/um4EXkia4HgyCiR5T6o1KAekPdxFBs
fmcLoPN+TmBO8ZpLGEqy1Vwf59ahDW7dQiLXTIAEiGCoXSWI9MvtHDZ2o4IBIDCC
ARwWegYDVR0TAQH/BAgwBgEB/wIBADAFBgNVHSMGDAWgBSqgKibD7tLpn7FAvRy
zSzxpsnZtzBABggrBgEFBQCBAQQ0MDIwMAYIKwYBBQUHMAKJGh0dHA6Ly9jcnQu
ZWlkcGtpLmVlL0VFR292Q0EyMDI1LmNydDA9BgNVHSAENjA0MDIGBFUdIAAwKjAo
BggrBgEFBQCcARYcaHR0cHM6Ly9yZXBvc210b3J5LmVpZHBraS51ZTA1BGNVHR8E
LjAsMCgqKKAhRiRodHRwOi8vY3JsLmVpZHBraS51ZS9FRUdvdKNBMjAyNS5jcmww
HQYDVR0OBBYEFJLAOLC4NhJo9crtZu5HKohtpo3oMA4GA1UdDwEB/wQEAWIBBjAK
BggqhkJOPQDAwNnADBKAjANipgLQgdM985dSFZfKvU9A7Ssz2YdmmUSZBxu01L7Q
XKzqa0ZDyXmf03NPLNAC6dICMBQiROZbLoPezO9LD1847UbeNx85hloLlzeWjqp
rY++Xj8FjCD1C9hnb1sVgJ3XAA==
-----END CERTIFICATE-----
```