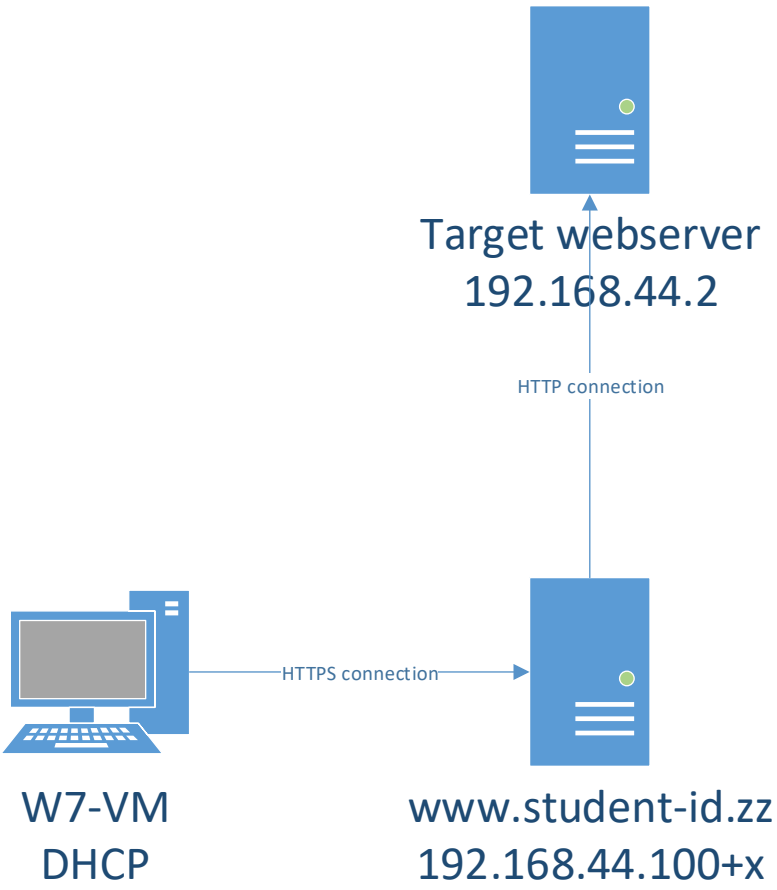
Lab3 – TLS Hardening

You can use this lab manual for your personal documentation. Use screenshots for your own documentation, there will be questions later on that may point to this lab manual. Take care to check if you need to collect some information from the lab for the answers.

\ at the end of the line is used to mark that the command needs to be on one line. Replace **student-id** with your own student-id and **x** or **y** as your VMs correct IP in the labs.

The labs use the following topology:



All VMs in this lab are in VirtualBox ***Bridged*** network. The machines that have static IP need to have an offset, check the topology image for reference. USE YOUR WIN7 WORKSTATION IP ADDRESS AS **x**.

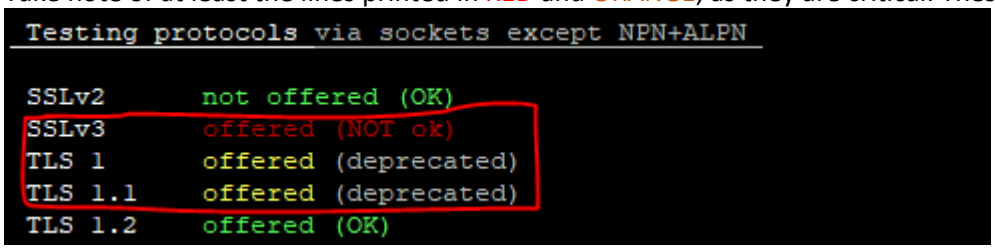
All templates for VMs can be found in \\ghost.labranet.jamk.fi\virtuaalikoneet\TTKS\

- **TestSSL.sh**

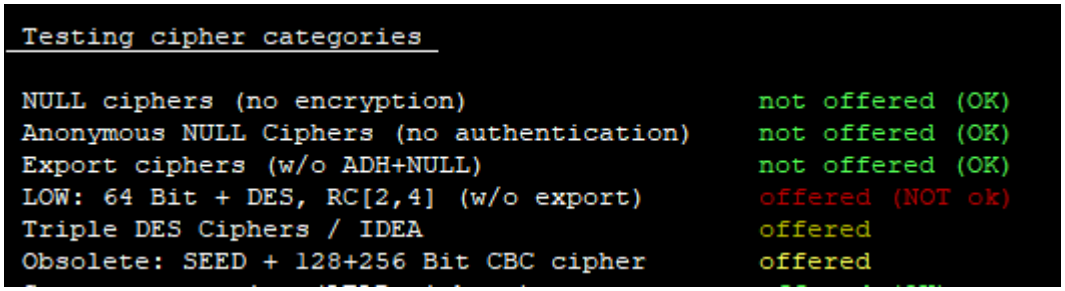
Before and after hardening, check results with testssl.sh (https://testssl.sh/). Download it to the proxy server and run against localhost:

```
yum install git
git clone --depth 1 https://github.com/drwetter/testssl.sh.git
cd testssl.sh
./testssl.sh https://www.student-id.zz/
```

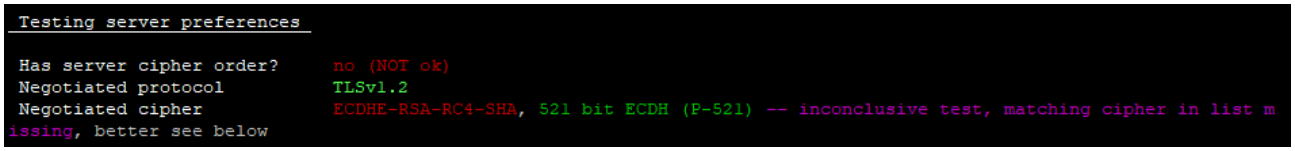
Take note of at least the lines printed in RED and ORANGE, as they are critical. These should be mitigated.



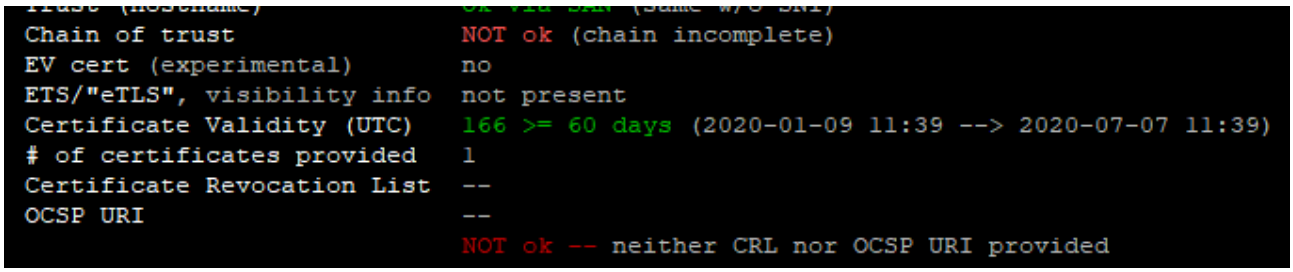
- SSl ja TLS1-1.1 ovat vanhentuneet, ja poistuneet käytöstä



-



- jos cipher order on päällä, käytetään parasta cipher salausta ensisijaisesti



- CA ei ole luotettujen certifikaattien joukossa

- poodle haavoittuvuus mahdollinen, jos joku ssl protokolla vielä käytössä.

- RC4 haavoittuvuus saattaa olla mahdollinen viel TLS1.1 prokollan ollessa käytössä.

- **Mozilla TLS**

Mozilla has a nice TLS configuration generator in https://mozilla.github.io/server-side-tls/ssl-config-generator/

Find out from your server:

- Your Apache version
    - Apache 2.4.6
- Your OpenSSL library version
    - OpenSSL 1.0.2k-fips

Using the Mozilla Generator, generate configuration for the server. Add this to your proxy.conf. Do not add OCSP configurations as we don't have a valid OCSP Responder for the CA Certificate.

```
<VirtualHost *:443>
ServerName www.m3227.zz

SSLEngine On
SSLCertificateFile /etc/pki/tls/certs/www.pem
SSLCertificateKeyFile /etc/pki/tls/private/www.key

Header always set Strict-Transport-Security "max-age=63072000"

ProxyPass / http://192.168.44.2/
ProxyPassReverse / http://192.168.44.2/
ProxyPreserveHost On
</VirtualHost>

<VirtualHost *:80>
   ServerName www.m3227.zz
   Redirect permanent / https://www.m3227.zz/
   RewriteEngine On
   RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>

<Location /admin>
ProxyPass !
Require all denied
</Location>

ProxyPass /local !
Alias /local /var/www/html

SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDH$
SSLHonorCipherOrder     off

<Location /local>
ProxyPass !
Require all denied
</Location>
```

After configuring these, check again with testssl.sh

- Kaikki muut korjaantuivat, paitsi kolme kohtaa, näistä yksi oli ocsp.

```
Has server cipher order?     no (NOT ok)
```

```
Chain of trust                NOT ok (chain incomplete)
```

- **Extra hardening**

If critical errors still occur when testing with testssl.sh, try to find out the causes for them and mitigate. These may vary depending on the current changes in updates, new vulnerabilities, etc. If unsure, ask the teacher.

- Lisätään conffeihin SSLHonorCipherOrder on

```
SSLEngine On
SSLCertificateFile /etc/pki/tls/certs/www.pem
SSLCertificateKeyFile /etc/pki/tls/private/www.key
SSLHonorCipherOrder on
```

```
Testing server preferences

Has server cipher order?     yes (OK)
Negotiated protocol          TLSv1.2
Negotiated cipher            ECDHE-RSA-AES128
Cipher order
    TLSv1.2:   ECDHE-RSA-AES128-GCM-SHA256 ECD
```

- Tuodaan CA hakemistoon /root/testssl.sh/etc/

```
[root@localhost.localdomain ca-trust-source]# scp root@192.168.44.85:ca/ca.pem /usr/share/pki/ca-trust-source/anchors/
```

```
 254  cp /usr/share/pki/ca-trust-source/anchors/ca.pem etc
```

```
Chain of trust                NOT ok: Apple (chain incomplete)
                              OK: ca
```