



Higher School of Communication of Tunis

User Guide

Smart House Monitoring System

Authored by:

Anas Ben Amor, Aymen Ktari

Anas.BenAmor@supcom.tn, Aymen.Ktari@supcom.tn

Supervised By:

Dr. Eng. Mohamed-Bécha Kaaniche

medbecha.kaaniche@supcom.tn

Academic Year:

2024-2025

1 Introduction

This document describes the steps to configure a WildFly server with an SSL certificate to secure the deployed application.

2 Server Preparation

Before starting the installation, ensure your system is up-to-date. Run the following commands to update packages:

```
sudo apt update && sudo apt upgrade
```

3 Java Installation

1. Download the JDK file with 'wget':

```
wget https://download.oracle.com/java/21/latest/jdk-21_linux-x64_bin.deb
```

2. Install the JDK:

```
sudo apt install ./jdk-21_linux-x64_bin.deb
```

3. Verify the Java installation:

```
java -version
```

4 WildFly Installation and Configuration

1. Download and extract WildFly:

```
wget https://github.com/wildfly/wildfly/releases/download/34.0.0.Final/wildfly-34.0.0.Final.tar.gz
tar -xvf wildfly-34.0.0.Final.tar.gz
sudo mv ./wildfly-34.0.0.Final /opt
```

2. Create a symbolic link to simplify access to WildFly:

```
sudo ln -s /opt/wildfly-34.0.0.Final /opt/wildfly
```

3. Add a dedicated user for WildFly:

```
sudo useradd -r -d /opt/wildfly -s /usr/sbin/nologin wildfly
```

4. Set up permissions and configuration files for WildFly:

```
cd /opt
sudo chown -RH wildfly:wildfly wildfly
sudo mkdir -p /etc/wildfly/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/
wildfly.conf /etc/wildfly/
sudo cp /opt/wildfly/standalone/configuration/
standalone.xml /opt/wildfly/standalone/
configuration/your-domain.me.xml
sudo vi /etc/wildfly/wildfly.conf
```

5. Modify the configuration file by changing WILDFLY_CONFIG to your-domain.me.xml.

6. Copy the startup scripts:

```
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/
launch.sh /opt/wildfly/bin/
sudo cp /opt/wildfly/docs/contrib/scripts/systemd/
wildfly.service /usr/lib/systemd/system/
```

7. Create the runtime directory for WildFly and set permissions:

```
sudo mkdir /var/run/wildfly/
sudo chown -RH wildfly:wildfly /var/run/wildfly/
```

8. Add TLS options to the Java configuration:

```
cd /opt/wildfly/bin
vi standalone.conf
```

Modification: Add `-Djdk.tls.server.enableStatusRequestExtension=true` to `JAVA_OPTS` in `standalone.conf`.

9. Modify the WildFly port configuration to listen on ports 80 and 443:

```
vi /opt/wildfly/configuration/your-domain.me.xml
```

Modification: At the end of the file, change 8080 to 80 and 8433 to 443.

10. Allow access to ports 80 and 443:

```
setcap CAP_NET_BIND_SERVICE=+eip /opt/wildfly/bin/
standalone.sh
setcap CAP_NET_BIND_SERVICE=+eip /opt/wildfly/bin/
launch.sh
setcap CAP_NET_BIND_SERVICE=+eip /usr/lib/jvm/jdk-21/
bin/java
```

5 Firewall Configuration

Open the necessary ports for the server:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

6 TLS Configuration

1. Install Certbot

- First, install Certbot to manage SSL certificates:

```
sudo apt install certbot
```

2. Generate SSL Certificate with Certbot (DNS Challenge)

- Use Certbot with the DNS challenge to generate a wildcard certificate. Certbot will provide a TXT record that needs to be added to your domain's DNS configuration:

```
sudo certbot certonly --manual --preferred-challenges
dns --manual-public-ip-logging-ok --must-staple -d "
*.smarthomecot.me." -d smarthomecot.me.
```

- When prompted by Certbot, copy the provided TXT record. Then, in your domain management portal:
 - Navigate to **Domain List - Manage - Advanced DNS**.
 - Add a **TXT Record** with the following details provided by Certbot:
 - * **Host:** `_acme-challenge`
 - * **Value:** (the TXT record provided by Certbot)
 - Save the record and wait for DNS propagation.
 - Verify the record is updated by checking the DNS record at the following link:
`https://toolbox.googleapps.com/apps/dig/#TXT/_acme-challenge.smarthomecot.me`.







<input type="checkbox"/> Type	Host	Value	TTL	
<input type="checkbox"/> A Record	@	72.145.3.153	Automatic	
<input type="checkbox"/> CNAME Record	www	smarthomecot.me.	30 min	
<input type="checkbox"/> CAA Record	@	0 issue "letsencrypt.org"	Automatic	
<input type="checkbox"/> CAA Record	@	0 issuewild "letsencrypt.org"	Automatic	
<input type="checkbox"/> CAA Record	@	0 iodef "mailto:aymen.ktari@supcom.tn"	Automatic	
<input type="checkbox"/> TXT Record	_acme-challenge	Ml5rMSpMJvFsYJFFUmy3raxuqEA1HQ8ILZFYYcIZnY	Automatic	

Figure 1: Namecheap Configuration

3. Navigate to the Certificates Directory

- Change to the directory where Certbot stores generated certificates:

```
cd /etc/letsencrypt/
```

4. Set Permissions for Live Directory

- Adjust permissions to access certificate files if required:

```
sudo chmod 777 live
```

5. Convert Certificates to PFX Format

- Move to your certificate directory:

```
cd live/smarthomecot.me
```

- Convert the .pem certificate files to .pfx format:

```
sudo openssl pkcs12 -export -out certificate.pfx -inkey  
privkey.pem -in cert.pem -certfile chain.pem
```

- **Password:** Enter an export password (e.g., changeit).

6. Import the PFX File into a Java Keystore

- Use keytool to import the .pfx file into a Java Keystore:

```
sudo keytool -importkeystore -srckeystore certificate.  
pfx -srcstoretype PKCS12 -srcstorepass 'changeit' -  
storepass 'changeit' -destkeystore smarthomecot.me.  
jks -deststorepass 'YourcertifPwd'
```

7. Move the Keystore to WildFly's Configuration

- Move the Java Keystore file to WildFly's configuration directory:

```
sudo mv ./smarthomecot.me.jks /opt/wildfly/standalone/
configuration/
```

8. Clean Up

- Remove the .pfx file if no longer needed:

```
sudo rm certificate.pfx
```

9. Configure Elytron for TLS in WildFly

- Add the keystore, key manager, and SSL context in WildFly using the following commands in Jboss-cli:

```
/subsystem=elytron/key-store=smarthomeKeyStore:add(path=
smarthomecot.me.jks,relative-to=jboss.server.config.
dir,credential-reference={clear-text="YourcertifPwd"
},type=JKS)

/subsystem=elytron/key-manager=smarthomeManager:add(key-
store=smarthomeKeyStore,credential-reference={clear-
text="YourcertifPwd"})

/subsystem=elytron/server-ssl-context=
smarthomeTLSContext:add(key-manager=smarthomeManager,
protocols=["TLSv1.3"],cipher-suite-names="
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:
TLS_AES_128_GCM_SHA256")
```

- Define and configure the HTTPS listener to use the new SSL context:

```
batch

/subsystem=undertow/server=default-server/https-listener
=https:undefine-attribute(name=security-realm)

/subsystem=undertow/server=default-server/https-listener
=https:write-attribute(name=ssl-context,value=
smarthomeTLSContext)

run-batch

reload
```

- Set the management interface to use the secure TLS context:

```
/core-service=management/management-interface=http-  
interface:write-attribute(name=ssl-context, value=  
smarthomeTLSContext)  
  
reload  
  
/core-service=management/management-interface=http-  
interface:write-attribute(name=secure-socket-binding,  
value=management-https)  
  
reload
```

10. Configure HSTS and HTTP-to-HTTPS Redirection

- Add HTTP Strict Transport Security (HSTS) and configure HTTP-to-HTTPS redirection:

```
/subsystem=undertow/configuration=filter/response-header  
=hsts:add(header-name=Strict-Transport-Security,  
header-value="max-age=63072000;␣includeSubDomains;␣  
preload")  
  
/subsystem=undertow/configuration=filter/rewrite=http-to-  
https:add(target="https://%v:443%U", redirect=true)  
  
/subsystem=undertow/server=default-server/host=default-  
host/filter-ref=hsts:add(predicate="equals(%p,443)")  
  
/subsystem=undertow/server=default-server/host=default-  
host/filter-ref=http-to-https:add(predicate="equals(%  
p,80)")
```

7 Starting and Verifying the WildFly Server

Start the WildFly server and check its status:

```
sudo systemctl daemon-reload  
sudo systemctl start wildfly  
sudo systemctl status wildfly
```

8 SSL Labs Test Result

- Below is an example image of the SSL Labs result, showing the configuration and security grade achieved.

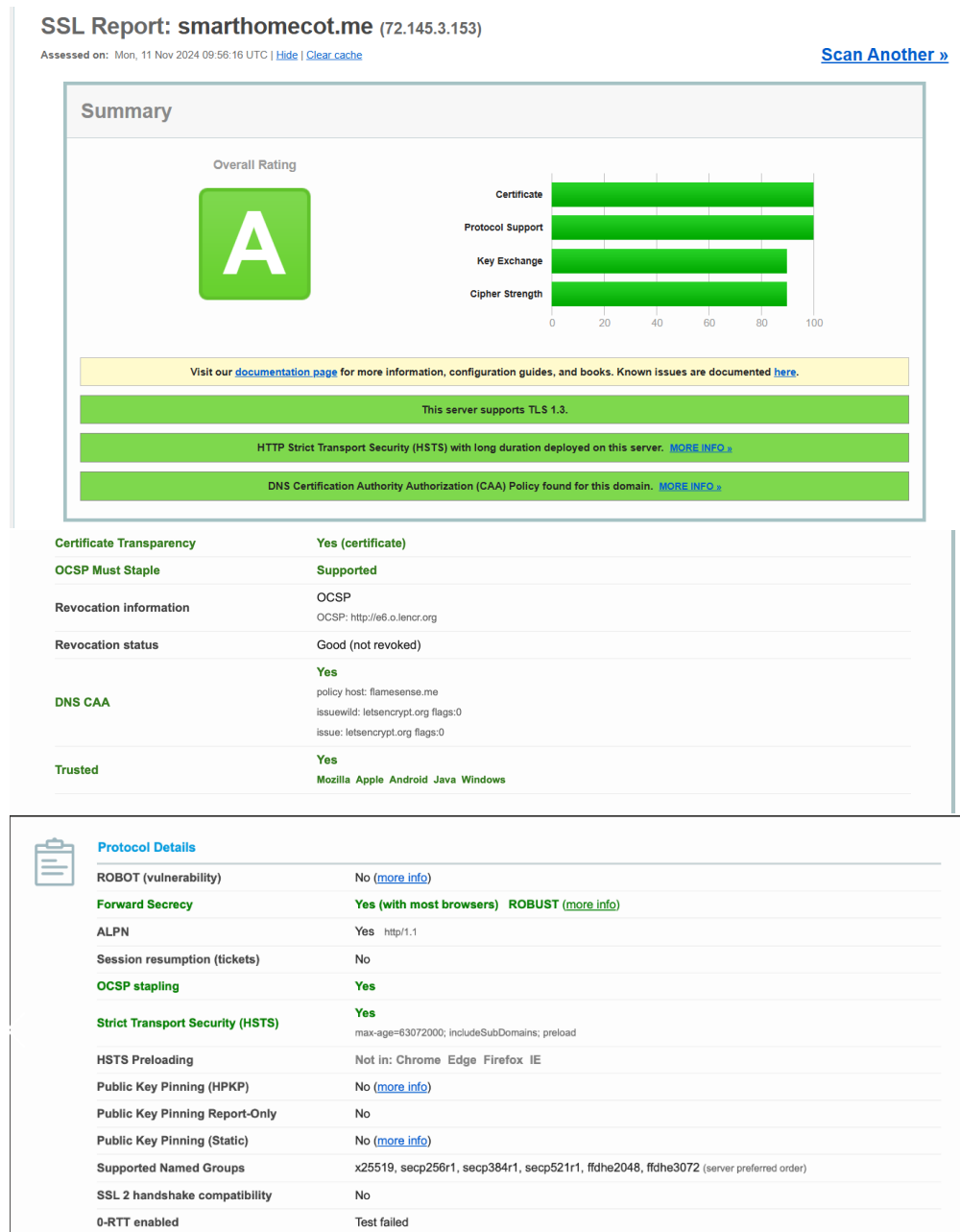


Figure 2: SSL Labs Result for TLS Configuration