Konfigurácia Wildfly AS pre SmartWeb

- Základná konfigurácia standalone.xml
- Odporúaná konfigurácia pre optimálny beh a zabezpeenie servera
- Volitená konfigurácia silného šifrovania pre HTTPS
- Volitená konfigurácia automatického presmerovania HTTP na HTTPS

Základná konfigurácia standalone.xml

Súbor standalone.xml je hlavný konfiguraný súbor aplikaného servera Wildfly a nachádza sa v adresári `/standalone/configuration`. SmartWeb server ma implementovanú funkciu automatickej konfigurácie pri deploymente aplikácie aplikaným serverom Wildfly. Z tohto dôvodu pre beh SmartWeb aplikácie ako takej nie je potrebné súbor standalone.xml editova. Editácia je nevyhnutná iba v prípade úpravy konfigurácie samotného aplikaného servera - napr. zmena portov, konfigurácia zabezpeenia komunikácie HTTPS a HTTP hlaviiek, zapnutie overovania cez klientské certifikáty, kompresia komunikácie at.

Pozor, editáciu konfiguraného súboru standalone.xml vykonávame zásadne pri vypnutom aplikanom serveri, z dôvodu že poas jeho behu si ju on sám spravuje a môže prepísa zmeny uložené cez editor.

Odporúaná konfigurácia pre optimálny beh a zabezpeenie servera

Pre optimálny beh SmartWeb servera a základné zabezpeenie cez HTTP hlaviky odporúame nasledovnú konfiguráciu:

Odporúané zmeny v standalone.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="urn:jboss:domain:4.2">
        ...
        <profile>
                <subsystem xmlns="urn:jboss:domain:logging:3.0">
            ...
            <!-- Vypnutie zbytonch info hlášok o ukonení websocket spojenia -->
            <logger category="org.cometd.websocket.server.WebSocketTransport$WebSocketScheduler$1">
                <level name="WARN"/>
            </logger>
            ...
        </subsystem>
                ...
                 <subsystem xmlns="urn:jboss:domain:undertow:3.1">
                        <server name="default-server">
                                <host name="default-host" alias="localhost">
                   <location name="/" handler="welcome-content"/>
                   <filter-ref name="gzipFilter" predicate="not min-content-size(450)"/>
                   <filter-ref name="Strict-Transport-Security-header"/>
                   <filter-ref name="Vary-header"/>
                   <filter-ref name="X-Frame-Options"/>
                   <filter-ref name="X-Content-Type-Options"/>
                   <filter-ref name="X-XSS-Protection"/>
                   <filter-ref name="Referrer-Policy"/>
                   <filter-ref name="Content-Security-Policy"/>
                </host>
            </server>
            ...
            <filters>
                <response-header name="Vary-header" header-name="Vary" header-value="Accept-Encoding"/>
                <response-header name="Strict-Transport-Security-header" header-name="Strict-Transport-
Security" header-value="max-age=31536000; includeSubDomains"/>
                <!-- Nastavenia pre Cross-Origin Resource Sharing (nepoužívané/zakázané) -->
                <response-header name="Access-Control-Allow-Origin" header-name="Access-Control-Allow-Origin"
header-value="*"/>
                <response-header name="Access-Control-Allow-Methods" header-name="Access-Control-Allow-Methods"
header-value="GET, POST, OPTIONS, PUT"/>
                <response-header name="Access-Control-Allow-Headers" header-name="Access-Control-Allow-Headers"
header-value="accept, authorization, content-type, x-requested-with"/>
                <response-header name="Access-Control-Allow-Credentials" header-name="Access-Control-Allow-
Credentials" header-value="true"/>
```

```
                <response-header name="Access-Control-Max-Age" header-name="Access-Control-Max-Age" header-
value="1"/>
                <!-- Zakázané vkladanie stránok do frame (starší spôsob) -->
                <response-header name="X-Frame-Options" header-name="X-Frame-Options" header-value="DENY"/>
                <!-- Vynútené použitie MIME typu nastaveného v HTTP hlavike -->
                <response-header name="X-Content-Type-Options" header-name="X-Content-Type-Options" header-
value="nosniff"/>
                <!-- Zakázané zobrazenie stránky, ak bol detekovaný cross-site scripting (XSS) útok -->
                <response-header name="X-XSS-Protection" header-name="X-XSS-Protection" header-value="1;
mode=block"/>
                <!-- Neodosielanie referrer informácií -->
                <response-header name="Referrer-Policy" header-name="Referrer-Policy" header-value="no-referrer"
/>
                <!-- Nastavenie bezpenostnej politiky obsahu:
                     - zakázané vkladanie do frame (nový spôsob)
                     - predvolene povolený zdroj obsahu z hostiteskej domény
                       - pre CSS štýly povolené aj zabezpeené https odkazy a inline
                       - pre súbory písiem povolené aj google písma
                       - pre skripty povolené inline aj evaluácia
                       - zakázané plugin objekty (flash a pod.)
                     - povolené pripájanie z ubovolnej lokality
                -->
                <response-header name="Content-Security-Policy" header-name="Content-Security-Policy" header-
value="frame-ancestors 'none'; default-src 'self'; style-src https: 'self' 'unsafe-inline'; font-src 'self'
https://themes.googleusercontent.com https://fonts.gstatic.com; script-src 'self' 'unsafe-inline' 'unsafe-
eval'; object-src 'none'; connect-src *"/>
                <gzip name="gzipFilter"/>
            </filters>
        </subsystem>
    </profile>
        ...
        <interfaces>
        ...
        <interface name="public">
            <!-- Nastavenie bind adresy na vsetky sietove interface, kvoli tomu aby bol Wildfly pristupny aj z
vonku -->
            <inet-address value="${jboss.bind.address:0.0.0.0}"/>
        </interface>
    </interfaces>


        <socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.
binding.port-offset:0}">
        ...
        <!-- Nastavenie portov pre HTTP a HTTPS na všeobecne používané hodnoty, pozor toto nemeni pri
inštalácii na Linuxe, vi kapitola o inštalácii na Linuxe nižšie -->
        <socket-binding name="http" port="${jboss.http.port:80}"/>
        <socket-binding name="https" port="${jboss.https.port:443}"/>
        ...
    </socket-binding-group>
</server>
```

Volitená konfigurácia silného šifrovania pre HTTPS

Nasledujúce zmeny v standalone.xml konfigurujú zapnutie silných šifier pre HTTPS protokol. Podmienkou je inštalácia Java Cryptography Extensions popísaná v kapitole Inštalácia JRE 1.8 a Git klienta.

Zmeny v standalone.xml pre HTTPS

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="urn:jboss:domain:4.2">
    ...
        <system-properties>
        <!-- Minimálna džka Diffie-Helman kúa -->
        <property name="jdk.tls.ephemeralDHKeySize" value="2048"/>
    </system-properties>

        <management>
        <security-realms>
            ...
            <!-- Security realn undertowTLSRealm je potrebné nastavi iba v prípade konfigurácie HTTPS, zárove
aj pre overovanie cez klientské certifikáty -->
            <security-realm name="undertowTLSRealm">
                <server-identities>
                    <ssl protocol="TLS">
                        <!-- Cesta/heslo ku keystore kde je uložený SSL certifikát pre HTTPS s definovaným
aliasom, napr. nb1mgre1.ipesoft-int.sk -->
                        <keystore path="server.jks" relative-to="jboss.server.config.dir" keystore-password="
secret" alias="nb1mgre1.ipesoft-int.sk" key-password="secret"/>
                    </ssl>
                </server-identities>
                <!-- as "authentication" je potrebné nastavi iba pre overovanie cez klientské certifikáty -->
                <authentication>
                    <truststore path="client-certificates.jks" relative-to="jboss.server.config.dir" keystore-
password="secret"/>
                </authentication>
            </security-realm>
        </security-realms>
        ...
    </management>

        <profile>
         ...
        <subsystem xmlns="urn:jboss:domain:undertow:3.1">
            ...
            <server name="default-server">
                ...
                <!-- !!! Atribút verify-client je potrebné nastavi na hodnotu REQUESTED v prípade ak je
potrebná autentifikácia cez klientské certifikáty, inak ho netreba meni -->
                <https-listener name="https" socket-binding="https" security-realm="undertowTLSRealm" verify-
client="NOT_REQUESTED"
                                enabled-protocols="TLSv1.2,TLSv1.1"
                                enabled-cipher-suites="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256"
                                enable-http2="true"/>
            </subsystem>
    </profile>

</server>
```

Uvedené zmeny konfigurácie majú nastavený zoznam povolených protokolov a šifier tak, aby bol bezpený, podporené sú len novšie prehliadae - IE11 a Android >= 4.4.3, Safari >= 7. V prípade potreby podpory ešte starších prehliadaov treba nastavit inak tieto atribúty:

```
enabled-protocols="TLSv1.2,TLSv1.1,TLSv1"
enabled-cipher-suites="TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256,TLS_DHE_DSS_WITH_AES_128_GCM_SHA256,TLS_DHE_DSS_WITH_AES_256_GCM_SHA384,
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256,TLS_DHE_DSS_WITH_AES_256_CBC_SHA256,TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA"
```

Volitená konfigurácia automatického presmerovania HTTP na HTTPS

V prípade že Wildfly AS je dostupný z vonku priamo cez ním otvorené porty a doménu je potrebná nasledovná konfigurácia:

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="urn:jboss:domain:4.2">
        ...
        <profile>
        ...
        <subsystem xmlns="urn:jboss:domain:undertow:3.1">
            ...
            <server name="default-server">
                                <!-- Atribút redirect-socket nastavi iba v prípade výhradnej komunikácie cez
HTTPS -->
                <http-listener name="default" socket-binding="http" redirect-socket="https"/>
                        </server>
            ...
                </subsystem>
    </profile>
</server>
```

V prípade že Wildfly AS je dostupný cez samostatný proxy server alebo IPTABLES rerouting je potrebná nasledovná konfigurácia:

```
<?xml version='1.0' encoding='UTF-8'?>
<server xmlns="urn:jboss:domain:4.2">
        ...
        <profile>
        ...
        <subsystem xmlns="urn:jboss:domain:undertow:3.1">
            ...
            <server name="default-server">
                                ...
                                <host name="default-host" alias="localhost">
                                        ...
                                                <!-- Atribút predicate treba nastavi na HTTP port definovaný v
poslednej asti standalone.xml-->
                                                <filter-ref name="http-to-https" predicate="equals(%p,8080)"/>
                                        ...
                                </host>
                    ...
            <filter-ref name="http-to-https" predicate="equals(%p,8080)"/>
                        </server>
            ...
                        <filters>
                                <!-- Atribút target treba nastavi na finálnu doménu a port, %U je placeholder
pre zvyšnú as otváranej url linky-->
                                <rewrite name="http-to-https" redirect="true" target="https://myhostname:8443%U"
/>
                        </filters>
                </subsystem>
    </profile>
</server>
```