|  |
| --- |
| ***Teme:***  *Generiranje in raba certifikatov /https/; OpenSSL in javanski keystore*  *http protokol, metode http protokola*  *realizacija http strežnikov\**  *REST Api in OAS3\** |

|  |
| --- |
| Delna specifikacija naloge (preliminarni test za izvedbo naloge in seznanjevanje z ekositemom/okoljem) je podano na : <https://gitlab.vegova.si/SerhioN/comptest05/tree/master/java>  Specifikacija API-ja: https://gitlab.vegova.si/SerhioN/comptest05/tree/master/openapi  Relevantni mapi sta torej:   * java (dva strežnika) * openapi (oas3 definicija)   REST – API (poenostavljen) definira:   * Strežnik, s katerim komuniciramo * EndPoint-e (URI-je, ki jih lahko naslovimo) * Parametre pri komunikaciji * Podatkovne modele podatkov (ki jih vrača, jih dajemo kot parametre) * Kdo/kdaj/do česa lahko dostopa (security) • ……   Postopki v zvezi s certifikati so spisani tudi v : <https://gitlab.vegova.si/SerhioN/comptest05/-/tree/master/certificate_infos>  Naloga sovpada z nalogo dijakov predmeta NSA, le da je njihova realizacija strežnika drugačna / apache/nginx/php/node, je pa v opisu njihove naloge dana specifikacija za rešitev nalog 5 in 6 te vaje. |

**Naloga 1**

Preskusi oba strežnika (http,https) op: oba bi morala delovati

a) http: ali se da nasloviti vse specificirane endpoint-e (path definirani v oas3)

b) enako kot http pod a), vendar je predhodno potrebno definirati certifikat. Ta certifikat generirajte z javanskimi orodji (glej datoteko postopek\_certifikat.txt). Pri njegovi generaciji uporabite pravo (vaše) ime in priimek, Organization Unit=rso2021, O=VegovaLjubljana, city=ljubljana, coutryCode=SI

Zaženite https strežnik in preverite dosegljivost strežnika z običajnim brskalnikom.

**Naloga 2**

V nadaljevanju bomo uporabili digitalna potrdila, generirana z OpenSSL in podpisana s strani podpisnika Vegova.CA.2. Podpisnik se nahaja na naslovu s končnico .6 .

***Opozorilo: v postopku kreiranja generirate tudi privatni ključ. Shranite ga, ker ga boste potrebovali tudi v naslednji vaji. Izgubljen ključ pomeni ponovitev celotnega postopka.***

Postopek, ki ga boste izvedli bo naslednji:

1. Generiranje zahtevka za digitalno potrdilo (csr)
2. Podpisovanje zahtevka na strežniku .6
3. Prevzem potrdila in potrdila podpisnika
4. Uvoz celotne verige potrdil v javanski keystore

Nastavitvena datoteka za zahtevek pri openssl se nahaja v repojih gitlab-a.

Vaše potrdilo mora v CN imeti specificirano domeno z vašim imenom in priimkom kot je to primer za Janeza : janez.jkranjski.si v SAN specificirate samo dva DNS polja : janez.jkranjski.si in [www.jkranjski.si](http://www.jkranjski.si), IP-ja ne definirajte. Za to boste morali ustrezno popraviti nastavitveno datoteko. Ker je domena enaka priimku in priimek ni unikaten niti na tej šoli, je v danem primeru ime domene na začetku razširjeno s predpono prve črke imena.

Shranite privaten ključ in potrdila (hm, .p12 datoteko, če vsebuje tudi privatni ključ vašega potrdila)

**Naloga 3**

Preskusite novo potrdilo v kombinaciji z https strežnikom iz naloge 1, ter s konfiguracijo enega izmed danih predstavitvenih strežnikov, npr. Apache. Prepričajte brskalnik, da bo verjel uporabljenemu potrdilu.

**--------------------------**

Poročilo naj vsebuje rezultat ukazov keytool -v -list -keystore imaVašegaKeystora (uvožena potrdila) in openssl x509 -in vašCertifikat.crt -text

Naloge spodaj so neobvezne …(neobvezno ni isto kot tega se ne dela …) …

**Naloga 4\***

Oba strežnika naj se odzivata na vse definirane endPoint-e. Odziv naj bo demonstrativen (da se vidi, da URL-ji delujejo; glej primer http strežnika). Definirajte in usposobite manjkajoče endpointe na obeh strežnikih.

V pomoč: editor.swagger.io - lahko uvozite openapi.json/yaml -> v UI dobite izrisane vse endpoint-e in podatkovne modele

**Naloga 5\* (glej dokumente NSA)**

V okviru https strežnika implementirajte API specificiran openapi/openapi.json ali openapi/apenapi.yaml datoteki. Pri tem morate implementirati tabelo podatkovne baze, ki jo omenjena datoteka specificira v razdelku Schemas, kjer imate opisan model za Strel. Endpoint-i API ja specificrani razdelka path (gen, get, get/st\_serije; pri tem je serija opredeljena kot tisti streli, ki imajo isti timestamp, gen pa vedno generira naključnih 10 strelov z istim timestampom; get vedno vrne vse strele v tabeli; get/st\_serije pa zgolj strele izbrane serije).

**Naloga 6\* (glej dokumente NSA)**

Realizirajte celoten CRUD za tabelo Strel! (dodajanje posameznega strela, brisanje, ažuriranje, vračanje vrednosti). Pri tem si definirajte svoj API.

\* so v vsakem primeru neobvezne