|  |
| --- |
| **Teme:**   1. izdelava kvalificiranega digitalnega potrdila, ki ga bomo v nadaljevanju potrebovali za avtentifikacijo, pri kriptiranju, elektronskemu podpisovanju   2. raba lokalnega razreševalnika spletnih naslovov (local resolver)  kvalificirana potrdila:  \* openssl  \* kreiranje samopodpisanega potrdila  \* kreiranje privatnega ključa, zahtevka za potrdilo  \* - pošiljanje v generacijo/podpis  \* - prevzemanje potrdila |

|  |
| --- |
| *Nalogi (naloga1 in naloga2) naj bosta dokončani do konca bloka vaj. Dosežek se ovrednoti in upada po času s faktorjem 0.2 po dnevih zaostanka(1, 2-3, 4-6). Nalogi pripadata bloku RSOv in bosta tam upoštevani kot del ocene. Samo poročilo pa je del vaj pri predmetu (RSOv).* |

|  |
| --- |
| Izhodišča in viri:   1. CompTest05, <https://gitlab.vegova.si/SerhioN/comptest05/>, dost. 4.03.2024   mapa certificate\_infos/config\_vaje.cfg   1. CA strežnik za podpisovanje certifikatskih zahtevkov in generiranje certifikatov: <https://193.2.190.6/certca/aindex.html> oz. [https:/brown.vegova.si/certca/aindex.html](https://brown.vegova.si/certca/aindex.html) 2. Repozitorij zgeneriranih certifikatov : <https://193.2.190.6/certca/certs> |

**Naloga 1 : Namestitev orodja openssl, test z izdelavo samopodpisanega potrdila**

1. Namestimo openssl: (<https://wiki.openssl.org/index.php/Binaries>) predlagam prvega na <https://slproweb.com/products/Win32OpenSSL.html> (MSI installer). Namestite kamorkoli, samo poglejte, kam ga boste namestili.
2. Odprite ukazno okno v bin mapo namestitve openssl in izvedite:

openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout mojcert.key -out mojcert.crt

med postopkom odgovorite na par vprašanj:

|  |
| --- |
| Country Name (2 letter code) [AU]:SI  State or Province Name (full name) [Some-State]:New York  Locality Name (eg, city) []:Ljubljana  Organization Name (eg, company) [Internet Widgits Pty Ltd]:Vegova Lj  Organizational Unit Name (eg, section) []:Lab Rac  **Common Name** (e.g. server FQDN or YOUR name) []:192.168.106.74  Email Address []:jst@vegova.lp |

1. Pri tem naj bo IP dejansko IP računalnika, za katerim sedite. Common name je obvezen element in je eden od elementov potrdila, ki se pri avtentifikaciji dejansko uporabijo za preverjanje.

Rezultat izvajanja sta digitalno potrdilo v datoteki mojcert.crt in privatni ključ v datoteki mojcert.key.

Skušajte odpreti potrdilo s pregledovalcem potrdil (dvojni klik na datoteki s potrdilom); snapshot (zaslonski posnetek) zavihka podrobnosti z izbranim poljem 'nosilec' naj bo del poročila.

Ah in da; odprite ključ z notepad++ in poglejte prvo vrstico. V primeru, da je specificirano 'encripted' je ključ zaščiten z geslom. Če tako potrdilo uporabljate pri storitvah s samodejnim zagonom, je to lahko moteče, zato: geslo lahko ostranite z openssl :

|  |
| --- |
| > openssl rsa -in mojcert.key -out mojcert\_nogesl.key |

**Naloga 2**

Strežnik navadno teče na registrirani domeni. Njegov spletni naslov je oblike **ime.priimek.ct**. Pri tem je 'ime' dejansko ime računalnika, 'priimek.ct' naslov omrežja (domena) /recimo: brown.vegova.si/. Certifikati, ki se uporabljajo v podjetju so vsi podpisani s strani podpisnika v podjetju, narejeni so namensko za posamezne storitve in omogočajo zaščito z vpisom kombinacije imena domene, morebitne poddomene, IP naslova in imena lokalnega računalnika v SAN polja generiranega certfikatnega zahtevka.

Certifikat bomo zato tokrat naredili drugače:

1. generiramo certifikatni zahtevek,
2. damo zahtevek v podpis in generiranje certifikata podpisniku,
3. namestimo od podpisnika pridobljen certifikat.

Opomba: organizacijsko ime za strežnik se bo za vsakega posameznika razlikovalo; tako bo pri osebi Janez Novak ime **janez.novak.ct** pri osebi Peter Klepec pač **peter.klepec.ct**.

1. Generiranje certifikatnega zahtevka (openssl)

Da ne bo potrebno vpisovati polj med generiranjem zahtevka, si pripravimo konfiguracijsko datoteko **config\_vaje.cfg** , 'dolpotegnemo' s prvega vira, podanega na uvodni strani te naloge (privzeto je to openssl.cnf) . V njej spremenimo zahtevna polja ( vrstice 37-40 in 60-65). V komentarjih (#) imate spisana vsa potrebna navodila. Nato zaženite generiranje certifikatnega zahtevka, kot je v spodnjem primeru za osebo janez kranjski:

openssl req -batch -new -config config\_vaje.cfg -out janez.kranjski.csr -keyout janez.kranjski.key

Varno shranite datoteko s ključem. Potrebovali jo boste še nekajkrat v nadaljevanju.

1. Generiranje certifikata s strani podpisnika

Na naslovu <http://193.2.190.6/> se nahaja začasen podpisnik. Podpisnik bo na voljo vsaj do 10.4.2024. Sledite navodilom, in po postopku prevzemite generirane certifikate.

1. V poročilo dodajte 'snapshot' zavihka podrobnosti potrdila z izbranim poljem 'alternativno ime nosilca'
2. Namestite potrdila podpisnikov v shrambo zaupanja vrednih certifikatov sistema windows (dvoklik na datoteko potrdila, nameščanje, samodejna izbira shrambe; če pri podpisniku ne gresta obe potrdili avtomatično, boste morali vsako potrdilo iz datoteke namestiti ločeno)
3. Namestite vaše potrdilo v shrambo potrdil, mesto naj se izbere avtomatično.

V shrambi morajo biti vsa 3 potrdila: CA v shrambi zaupanja vrednih, podpisnik v zaupanja vrednih vmesnih overoviteljih, vaše v … Upravitelj potrdil na windows lahko zaženete tudi iz konzolnega okna z ukazom certmgr.

**Naloga 3**

Vaš računalniški sistem se mora odzivati na poljuden spletni naslov specificiran v potrdilu, kot tudi na internetni naslov s potrdila;

1. Preskusite odziv : ping IPv4 naslov vašega računalnika
2. Preskusite odziv: ping na naslov v obliki janez.kranjski.ct:

Slednji se ne odziva, ker ime janez.kranjski.ct pač ni registrirano in ga DNS storitev ne uspe pretvoriti v IP naslov. Ker vpliva na zapise DNS navadno nimamo, popravimo lokalni 'resolver' oz. razreševalec internetnih imen. Na windows gre to kot:

Odpremo windows\system32\drivers\etc\hosts in dopišemo ustrezno preslikavo (na konec datoteke). Za shranjevanje potrebujete admin pravice (npr. notepad++ zna sam preklopiti v admin način):

# za janez.kranjski.ct

192.168.106.110 janez.kranjski.ct #preslikava iz IP v ime in obratno (kot dela DNS)

V primeru, da ima vaš računalnik gornji IPv4 … kot je dano.

Preskusimo s : ping janez.kranjski.ct na lokalnem računalniku

'snapshot' rezultata izvedbe ping dodate v poročilo.

**Naloga 4**

1. Preskusite oba strežnika (http in https) dana na naslovu:

https://gitlab.vegova.si/SerhioN/comptest05/-/tree/master/java

za https strežnik boste potrebovali certifikat (potrdilo). V prvi aproksimaciji generirate samopodpisanega v javanski keystore, kot je v komentarju na začetku datoteke z izvorno kodo. Preverite **geslo** in **ime keystora** (morata ustrezati tistim v kodi)

keystore je datoteka, v tem primer s končnico jks. Java lahko uporablja lastno shrambo za potrdila (to je to, v tem, primeru, keystore) ali pa tisto od Windows. V poročilu zgolj specificirajte, ali strežnika delujeta in na katere endpointe se (posamezna) odzivata.

1. Uporabite 'svoj' openssl x.509 certifikat iz naloge 2 te vaje z java keystore

**b.1)** Pretvori x509 certificate v PFX obliko z OpenSSL

openssl pkcs12 -export -out marko.k.si.pfx -in marko.k.si.crt -inkey marko.k.si.key -name marko.k.si -CAfile rootCA.crt -noiter

(rezultat je .pfx datoteka, geslo MORATE podate, zapomnite si ga)

Sledita alternativi glede na to katero vrsto keystora želite uporabiti – v splošnem je pks12 enostavneje …

**b.2a)** Uvozi pfx v keystore vrste JKS z javanskim orodjem keytool

keytool -importkeystore -srckeystore marko.k.si.pfx -srcstoretype pkcs12 -destkeystore moj\_keystore.jks -deststoretype JKS

(če še nimate keystora, vam ga ustvari: datoteka moj\_keystore.jks - pazi: vnašaš gesla za PFX z koraka b.1) in za keystore/če se na novo kreira; 2x/

**b.2b)** Uvozi v keystore vrste PKCS12 z javanskim orodjem keytool; pkcs12 je industrjski standard

keytool -importkeystore -srckeystore marko.k.si.pfx -srcstoretype pkcs12 -destkeystore moj\_keystore.p12 -deststoretype pkcs12

(če še nimate keystora, vam ga ustvari: datoteka moj\_keystore.p12 - pazi: vnašaš gesla/

**c) mimogre**: v .p12 lahko exportirate direktno iz hrambe potrdil sistema windows ()

**d)** če CA certifikata ni v .p12 ali .jks naredite iz certifikatov skupek (bundle, tako, da enostavno datoteke skopirate v eno, CA naj bo na dnu, ker sta v obstoječem bundlu že dva certifikata s CA na dnu, ga uporabimo:

copy marko.k.si.crt+ [ca-chain.cert.crt](https://193.2.190.6/certca/certs/ca-chain.cert.crt) **ca\_bundle.crt** in v b1) uporabite -CAfile ca\_bundle.crt ....

v poročilo oddajte :

a) dir mape v kateri sta oba keystora

b) rezultat izpisa :

keytool -list -v -keystore moj\_keystore.jks

c) rezultat izpisa :

keytool -list -v -keystore moj\_keystore.p12

----------------------------------------------------------------

Nalogi 1 in 2 morata biti dokončani na uri, ocena gre k oceni RSOv; zakasnjene izvedbe se ocenijo z 20% odbitkom v zakasnitvi do 1 dneva, naslednjih 20% za naslednja 2 dneva, še 20% za dneve 4-6.