Mérési útmutató a

“Publikus Kulcsú Infrastruktúra”

című méréshez

2016. február

A mérést kidolgozta:

Paulik Tamás

Microsec zrt.

BME, CrySys Adat- és Rendszerbiztonsági Laboratórium

A mérés célja

A mérés célja, hogy a Hallgatók megismerkedjenek a Publikus Kulcsú Infrastruktúra legfontosabb gyakorlati alkalmazásaival és gyakorlati tudásra tegyenek szert az alábbi témakörökben:

- SSL kapcsolatok fogadására alkalmas webszerverek felállítása és konfigurálása
- Tanúsítvány alapú autentikáció fogadására alkalmas webszerverek felállítása, az autentikált tanúsítvány adatainak továbbítása a háttéralkalmazások felé
- Elektronikus aláírás készítése automatizált szerveroldali és kliensoldali eszközökkel
- A különböző aláírás típusok közötti különbségek megismerése

Mérési környezet bemutatása

A mérés az interneten szabadon elérhető szoftverekkel, valamint a Microsec zrt. Hitelesítés-szolgáltatója által biztosított eszközökkel történik. A mérés során az alábbi eszközök kerülnek felhasználásra:

Szoftveres PKI eszközök

- e-Szignó kártyakezelő: A mérés során használt intelligens kártya kezelésére használt szoftver
- e-Szignó kliens: A kliens oldali aláírási feladatok során használt elektronikus aláíró szoftver
- e-Szignó parancssoros eszköz: A szerver oldali aláírási feladatok során használt elektronikus aláíró szoftver
- Gemalto CC intelligens kártya driver: A mérés során használt intelligens kártyák meghajtóprogramja
- openssl: Az alapvető PKI műveletek elvégzésére

PKI hardware-ek

- Gemalto intelligens kártya
    - PIN kód: 123456

Méréshez szükséges egyéb szoftverek

- Adobe Reader: A PDF aláírások vizsgálatához és készítéséhez
- Apache httpd: A mérésekben ezt a webszerver implementációt használjuk
- Cygwin terminal: Ebben kerül futtatásra az openssl
- PHP interpreter: A webszerver mögötti szoftverkörnyezet ebben kerül megvalósításra

Egyéb felhasználói szoftverek

A mérés során használt szoftverkörnyezet ezen felül még az alábbi szoftvereket tartalmazza:

- Chrome böngésző
- Notepad++
- Total Commander

Beugró kérdések

A Publikus Kulcsú Infrastruktúra laboron történő részvétel előfeltétele a sikeres beugró teljesítése. A beugróban az alább felsorolt egyszerű és komplex kérdésekből kell megválaszolni néhányat.

A beugróban 6 egyszerű kérdésre kell válaszolni, darabonként egy pontért, valamint egy komplexre négy pontért. A beugró akkor sikeres, ha a megszerezhető 10 pontból a Hallgató legalább 6 pontot szerez.

A kérdésekre a helyes válaszok kivétel nélkül megtalálhatóak a Nagy e-Szignó Könyvben, amely letölthető az alábbi címről: http://static.e-szigno.hu/anyagok/nagy_e-szigno_konyv.pdf

Mivel elsődleges célunk, hogy a laborra minden Hallgató felkészülten, a szükséges PKI alapokkal felvértezve érkezzen, a felkészülés során felmerülő kérdéseket lehetőség van közvetlenül nekünk - a tananyagot kidolgozó Microsec zrt. PKI szakértőinek - feltenni. Arra kérünk mindenkit, hogy amennyiben elakadna, vagy egyes témakörökben további kérdései vannak, lépjen velünk kapcsolatba az oktatas@microsec.hu címen, ahol minden kérdésre örömmel válaszolunk.

Egyszerű kérdések

Ezekre a kérdésekre csak röviden, tömören a lényegre koncentrálva kell választ adni legfeljebb 2-3 mondatnyi terjedelemben.

1) Mi a legfontosabb előnye az aszimmetrikus kulcsú kriptográfiának a szimmetrikus kulcsúval szemben? És fordítva?
2) Mi a különbség az elliptikus görbe alapú (ECC) aszimmetrikus kulcsú kriptográfia és az RSA alapú aszimmetrikus kulcsú kriptográfia között?
3) Hogyan áll össze egy hatékony dokumentumtitkosítási megoldás a szimmetrikus és aszimmetrikus kriptográfia eszközeinek együttes alkalmazásával?
4) Mi az alapvető különbség a PKI és a PGP között? Melyik hogyan igazolja a személy és az ő nyilvános kulcsának összetartozását?
5) Mit nevezünk tanúsítványnak?
6) Funkció szerint milyen csoportokba sorolhatjuk a végfelhasználói tanúsítványokat?
7) Sorolj fel legalább ötöt a tanúsítványt felépítő fontosabb X.509 mezők közül!
8) Milyen módokon teheti közzé egy hitelesítés szolgáltató a visszavonási információkat?
9) Mi egy tanúsítvány életciklusa? (Ábra a Nagy e-Szignó Könyv 140. oldalán)
10) Milyen adatokat tartalmaz, mi célt szolgál egy PKCS#10 kérelem?
11) Miért kell az XML állományokat XMLDSIG aláírás (és lenyomatképzés) előtt kanonizálni?
12) Mi az a tanúsítványlánc és miért építünk tanúsítványláncot?
13) Miért használunk időbélyeget? Mit igazol az időbélyeg? Miért fontos időbélyeggel ellátott elektronikus aláírást használni?
14) Mi az a kivárási idő? Miért lehet rá szükség? Milyen esetben nincs szükség kivárási időre?
15) Az aláírás hosszú távú érvényességének megőrzéséhez miért van szükség felülidőbélyegzésre? Mit biztosítanak a további időbélyegek?
16) Mitől válhat egy érvényes aláírás ellenőrizhetetlenné?
17) Mi különböztet meg egy Extended Validation (EV) webszerver tanúsítványt egy Domain Validated, vagy Organization Validated webszerver tanúsítványtól?
18) Mi a különbség a wildcard és az UCC webszerver tanúsítványok között?
19) Az alábbi domainek közül melyikre illeszkedik a *.e-szigno.hu wildcard webszerver tanúsítvány?
    - www.e.szigno.hu
    - e-szigno.hu
    - www.e-szigno.hu
    - www.crysys.hu

- w.w.w.e-szigno.hu
- srv.e-szigno.hu

20) Sorolj fel hármat a PKI lehetséges gyakorlati alkalmazási területei közül. (Nagy e-Szignó Könyv 12.3 fejezet)
21) Mi az a Biztonságos Aláírás Létrehozó Eszköz (BALE-SSCD)?

Komplex kérdések

Ezekre a kérdésekre bővebb, kifejtendő választ kell adni, amely bizonyítja, hogy a Hallgató ismeri és érti a bemutatásra kerülő folyamatot. A válasz részletességének nem kell megegyeznie a Nagy e-Szignó Könyvben található tartalomével, a válaszadáskor koncentráljunk a lényegre a folyamat szempontjából kiemelten fontos lépésekre.

1) Mi az aláírás elkészítésének folyamata?
2) Mik az időbélyeges aláírás ellenőrzés lépései? („Aláírás ellenőrzése, -T” fejezet a Nagy e-Szignó Könyv 260. oldalától)
3) Mik egy SSL handshake lépései, kétoldali autentikáció esetén és milyen adatok cserélnek gazdát a handshake során?

Mérési feladatok

Webszerver tanúsítványok igénylése és kibocsátása

Ebben a mérési feladatban fel fogunk állítani egy kezdetleges hitelesítés szolgáltatói hierarchiát, valamint kulcsot és hozzá tartozó PKCS#10 formátumú tanúsítvány kiállítási kérelmet fogunk generálni. A felállított hierarchia segítségével kibocsátunk egy webszerver tanúsítványt, majd felkonfiguráljuk a webszerverünket, hogy a tanúsítvány használatával fogadjon SSL kapcsolatokat.

Önaláírt CA tanúsítvány kibocsátása

Kulcs generálása a CA számára

Ahhoz, hogy a mérés során felállhasson egy olyan webszerver, amelynek saját a mérés során kiadott webszerver tanúsítványa van, először szükséges egy hitelesítés szolgáltató gyökértanúsítvány kibocsátása, amely aláírhatja majd a webszerver tanúsítványt.

Indítsuk el a Cymkdir gwin alkalmazást és a *meres* felhasználó home könyvtárában hozzunk létre egy új mappát pkilabor néven. Ebben a mappában hozzunk létre két másik mappát certificate_authority és webserver_admin néven. A továbbiakban ezeken belül dolgozzunk.

A Cygwin által mutatott */home/meres* mappa a Windows fájlrendszerben a *c:\cygwin64\home\meres* elérési úton található meg (ha a *meres* mappa nem létezik, azt is hozzuk létre). A certificate_authority mappában fogjuk elvégezni azokat a műveleteket, amelyeket egy hitelesítés-szolgáltatónak el kell végeznie, míg a webserver_admin mappában a webszerver adminisztrátorának munkafolyamatát követhetjük végig.

Lépjünk be a certificate_authority mappába. Generáljunk egy önaláírt tanúsítványt a hitelesítés szolgáltató számára! Ehhez használjuk az **openssl req** parancsot a megfelelő paraméterezéssel. A parancssori paraméterek az alábbiak legyenek:

- A kimenet x509 formátumú legyen.
- Az újonnan generált kulcs egy 2048 bites RSA kulcspár legyen.
- A tanúsítvány aláírási lenyomatképző algoritmusa sha256 legyen.
- A tanúsítvány érvényessége 365 nap legyen.
- Kimenetként kerüljön elmentésre a generált privát kulcs ca.key és a tanúsítvány ca.crt néven.

A parancssori paraméterekkel kapcsolatban segítséget a Cygwin terminálban az alábbi paranccsal kaphatunk: **openssl req ?.** A generálás során megadandó adatokat az alábbiak szerint válasszuk:

- A privát kulcsot védő jelszó legyen pkilabor
- Country Name: HU
- State or Province: Budapest
- Locality: Budapest
- Organization: Crysys Lab
- Organization Unit: PKI Labor
- Common Name: PKI Labor CA

- Email Address: ezt hagyjuk üresen

Önaláírt CA tanúsítvány vizsgálata

Vizsgáljuk meg a most kibocsátott tanúsítványt! Ehhez használjuk az openssl x509 parancsot: Segítséget újfent a ? paraméter megadásával kérhetünk. A paramétereket az alábbiak szerint válasszuk:

- Bemeneti formátum: PEM
- Bemeneti fájl: ca.crt
- Text formátumú kimenetet kérjünk
- Ne adjunk meg kimeneti fájlt. (ehhez meg kell adnunk a -noout paramétert)

Láthatjuk, hogy az Issuer és a Subject mezők tartalma azonos, láthatjuk a tanúsítvány érvényességi idejét. Az x509v3 extensions szekcióban láthatjuk, hogy a Subject Key Identifier és az Authority Key Identifier azonos, azaz önaláírt tanúsítvánnyal van dolgunk. A Basic Constraints mező mutatja, hogy egy hitelesítés szolgáltatóhoz tartozó tanúsítványt hoztunk létre.

Webszerver tanúsítvány kibocsátása

Ahhoz, hogy a felállítandó webszerverünk számára egy hitelesítés szolgáltató webszerver tanúsítványt bocsáthasson ki először szükséges generálnunk egy kulcspárt, majd a kulcspár alapján egy CSR-t (Certificate Signing Request). Ezt a CSR-t kell eljuttatnunk a hitelesítés szolgáltatónak, aki az ebben található adatok alapján kibocsátja a tanúsítványt.

Kulcs és CSR generálás

Lépjünk át a webserver_admin mappába. Generáljunk kulcspárt és CSR-t az openssl req parancs használatával. A paramétereket az alábbiak szerint válasszuk:

- Válasszuk az új request generálását.
- Válasszuk az új kulcs generálását. A kulcs legyen 2048 bites RSA kulcs.
- A kérelem a pkilabor.crysys.hu.csr fájlba kerüljön
- A kimeneti kulcs a pkilabor.crysys.hu.key fájlba kerüljön

A generálás során megadandó adatokat az alábbiak szerint válasszuk:

- A privát kulcsot védő jelszó legyen pkilabor
- Country Name: HU
- State or Province: Budapest
- Locality: Budapest
- Organization: Crysys Lab
- Organization Unit: PKI Labor
- Common Name: pkilabor.crysys.hu
- Email Address: ezt hagyjuk üresen
- Challenge password: pkilabor_challenge

- Company name: Crysys Lab

CSR vizsgálat

Másoljuk át a keletkezett pkilabor.crysys.hu.csr állományt a certificate_authority mappába, majd lépjünk át oda, hiszen a CSR ellenőrzését a hitelesítés szolgáltató végzi el. Vizsgáljuk meg és ellenőrizzük le a létrejött CSR-t! Ehhez használjuk az openssl req parancsot. A paramétereket az alábbiak szerint válasszuk:

- Kérjünk szöveges kimenetet
- Ne kérjünk fájl kimenetet (-noout)
- Kérjük a CSR-en található aláírás ellenőrzését
- Bemenetként adjuk meg a pkilabor.crysys.hu.csr fájlt

A kimenetben visszaköszönnek az általunk a kérés generálásakor megadott paraméterek. A kimenet tetején a 'verify OK' üzenet mutatja, hogy a CSR-en található aláírás nem sérült. A CSR-be továbbá belekerül a challenge jelszó is, amelyet korábban megadtunk. (Az openssl sajnos ebben a nézetben nem jeleníti meg a jelszó értékét.) Ez a mező biztosít lehetőséget egy hitelesítés szolgáltató számára, hogy egy ügyfélnek challenge jelszót kibocsátva később bármikor befogadhasson az ügyféltől CSR-t, amelyet a jelszó ismerete alapján autentikál.

Webszerver tanúsítvány kibocsátása

A CSR ellenőrzését követően a hitelesítés szolgáltatónak minden rendelkezésére áll, ami alapján kibocsáthatja a tanúsítványt.

Bocsássunk ki a webszerverünk számára tanúsítványt a CSR-ben található adatokra alapozva az openssl x509 parancsot, az alábbi paraméterekkel:

- Az üzemmód -req legyen (CSR befogadására, aláírásra és tanúsítvány kibocsátásra használjuk)
- A tanúsítvány érvényessége legyen 20 nap
- A bemeneti fájl a pkilabor.crysys.hu.csr legyen
- A használt CA tanúsítvány a ca.crt legyen
- A használt CA kulcs a ca.key legyen
- A tanúsítvány sorozatszáma legyen 00001
- A kimenet a pkilabor.crysys.hu.crt fájl legyen

A folyamat során szükség lesz a CA kulcsvédő jelszavára (pkilabor). Mivel a kibocsátás során nem adtunk meg kimeneti formátumot, a tanúsítvány PEM formátumú lesz.

Webszerver tanúsítvány vizsgálata

Vizsgáljuk meg a kibocsátott tanúsítvány tartalmát a már használt openssl x509 paranccsal.

Apache http szerver konfigurálása https kapcsolatokhoz

A frissen kiadott webszerver tanúsítványunk segítségével már felkonfigurálhatunk egy webszervert, amely fogad https kapcsolatokat.

Első lépésként a kiadott pkilabor.crysys.hu.crt állományt másoljuk át a webserver_admin mappába és lépjünk is át oda. („A hitelesítés-szolgáltató eljuttatta a kibocsátott tanúsítványt az igénylőnek.”)

Hosts file módosítás

Vegyük fel az alábbi sort a C:\Windows\System32\drivers\etc\hosts fájlba:

127.0.0.1 pkilabor.crysys.hu # Crysys PKI labor re-route

Webszerver indítás

(A mérés során egy valódi üzemeltetési környezettől eltérően a XAMPP programcsomagot használjuk a webszerver menedzsmentjére. A XAMPP csomag lehetővé teszi számunkra, hogy kényelmesebben kezeljük a webszervert és a mérés szempontjából lényeges feladatokra koncentráljunk)

Indítsuk el a XAMPP Control Panelt: C:\xampp\xampp-control. Kattintsunk az Apache sorában található Start gombra.

Miután az Apache elindult, indítsuk el a böngészőt és ellenőrizzük le, hogy elindult-e a "localhost"-on a webszerver. Ha minden rendben van, akkor a "Welcome to XAMPP for Windows" oldal fogad minket.

Ellenőrizzük le, hogy a hosts fájlban található módosításaink érvényre jutottak-e. A címsor használatával navigáljunk a http://pkilabor.crysys.hu címre. Ha az átirányítás működik, akkor megint a XAMPP nyitóképernyőnek kell megjelennie.

SSL engedélyezés és konfiguráció

Ahhoz, hogy a webszerverünket SSL kapcsolatokhoz felkonfiguráljuk, meg kell adnunk a most kiadott tanúsítványt és kulcsot az Apache konfigurációs állományában. Mivel Windows platformon az apache httpd nem támogatja a jelszóval védett privát kulcsok kezelését, ezért először el kell távolítanunk a kulcsot védő jelszót.

Cygwinben navigáljunk a /home/meres/pkilabor/webserver_admin mappába és távolítsuk el a jelszót a privátkulcsról. Ehhez használjuk az alábbi parancsot:

openssl rsa -in pkilabor.crysys.hu.key -out pkilabor.crysys.hu_unprotected.key

---

Bár az openssl dokumentációjából közvetlenül nem olvasható ki, ez a parancs eltávolítja a jelszót a kulcsfájlról. Az –in paraméter hatására a kulcs beolvasásra kerül (jelszóbekéréssel együtt), majd az -out paraméter hatására kiírásra kerül a lemezre. Az out paraméter viszont alapértelmezetten jelszavas

védelem nélkül írja ki a lemezre a kulcsokat. Ha jelszóval védve szerettük volna kiírni, akkor meg kellett volna adnunk a titkosító paraméterek valamelyikét.

Navigáljunk a C:\xampp\apache\conf mappába és nyissuk meg a httpd.conf állományt. Ez tartalmazza apache szerverünk általános konfigurációját. Ellenőrizzük le, hogy az SSL modul (mod_ssl) be van-e kapcsolva.

Nyissuk meg az extra/httpd-ssl.conf állományt. Ebben a fájlban találhatóak az apache szerverünk SSL konfigurációját. Keressük meg az alábbi paramétereket és állítsuk be értelemszerűen.

- ServerName: pkilabor.crysys.hu:443
- SSLCertificateFile: Az előző lépésekben kiadott SSL szerver tanúsítvány
- SSLCertificateKeyFile: Az előző lépésekben kiadott SSL szerver kulcsfájlja (jelszó nélküli)
- SSLCertificateChainFile: Az SSL szerver tanúsítványunk lánca (Jelen esetben a CA tanúsítvány)

Indítsuk újra az apache szolgáltatást és nyissuk meg Chrome böngészőben a https://pkilabor.crysys.hu címet!

A böngésző figyelmeztetni fog minket, hogy a kapcsolat nem biztonságos. A felületen az alábbi hibaüzenet jelenik meg: NET::ERR_CERT_AUTHORITY_INVALID Ez figyelmeztet minket arra, hogy a tanúsítvány elutasításának oka az, hogy ismeretlen a kibocsátó CA.

Kattintsunk a Speciális gombra a felület bal alsó sarkában. Válasszuk a Tovább... opciót.

Tanúsítvány telepítése a böngésző tanúsítványtárába

Ahhoz, hogy a böngészőnk elfogadja, az általunk kibocsátott tanúsítványt fel kell vennünk a kibocsátó CA-t a megbízható tanúsítványok közé. Ezt kizárólag akkor tesszük meg, ha ismerjük és megbízunk a CA-ban, ami most teljesül. Böngészés közben általában ez a feltétel nem teljesül, ezért ilyen hiba esetén nem telepítjük a CA tanúsítványát.

Telepítsük az általunk kiadott CA tanúsítványt a Windows tanúsítványtárába. Ehhez navigáljunk az alábbi mappába: C:\cygwin64\home\pki\pkilabor\certificate_authority, kattintsunk duplán a ca.crt fájlra és kövessük a varázsló lépéseit. A tanúsítványtároló opciónál válasszuk ki a „Megbízható legfelső szintű hitelesítés-szolgáltatók” tárolót.

Indítsuk újra a böngészőt és térjünk vissza az általunk konfigurált weboldalra.

Látható, hogy a böngésző innentől elfogadja az általunk kiadott tanúsítványt.

Tanúsítvány alapú autentikáció bekapcsolása

Amennyiben a webszerver rendelkezik SSL tanúsítvánnyal, lehetőségünk nyílik arra, hogy a csatlakozó klienseket is PKI alapon, autentikációs tanúsítványuk segítségével is azonosíthassuk. Az apache

felkonfigurálható úgy, hogy csak a megadott hitelesítés szolgáltatók által kiadott tanúsítványokat fogadja el, illetve képes CRL alapon elvégezni a tanúsítványok visszavonás-ellenőrzését.

A kártya és kártyaolvasó működőképességének ellenőrzése

Mielőtt beállítjuk a tanúsítvány alapú autentikációt, célszerű ellenőrizni, hogy a számítógéphez csatlakoztatott intelligenskártya és kártyaolvasó működőképes.

Indítsuk el az e-Szignó Kártyakezelő alkalmazást! A helyes működés ellenőrzéséhez végezzük el az alábbiakat:

- A nyitólapon az Olvasó és Kártya mezőknek kitöltött értékkel kell rendelkezniük
- A Tartalom lapon látnunk kell egy aláíró tanúsítványt és egy autentikációs tanúsítványt
- A Bejelentkezésre kattintva és a PIN kódot megadva be kell tudnunk jelentkezni. A sikeres bejelentkezést az jelzi, hogy a Bejelentkezés gombot a Kijelentkezés váltja fel

A sikeres tesztelést követően jelentkezzünk ki a kártyáról.

Elfogadott tanúsítványok tárának építése

A mérés során használt intelligens kártyán található autentikációs tanúsítványt az e-Szignó Teszt CA adta ki. Ahhoz, hogy a webszerverünket felkonfigurálhassuk a tanúsítvány befogadására, először be kell szereznünk a hitelesítés szolgáltató tanúsítványát.

Töltsük le a szükséges szolgáltatói tanúsítványokat! Ehhez kövessük az alábbi lépéseket:

- Látogassunk el a https://srv.e-szigno.hu oldalra
- A jobb oldali menüben válasszuk a "Szolgáltatói tanúsítványok" menüpontot
- Görgessünk le a lap aljára, majd Jobb klikk/Link mentése másként opcióval mentsük le a C:\cygwin64\home\meres\pkilabor\webserver_admin mappába a e-Szignó Test CA3 tanúsítványt és a Microsec e-Szigno Test Root CA 2008 gyökértanúsítványt.

A CA tanúsítványokat az Apache PEM formátumban várja el, amit most letöltöttünk viszont DER formátumban van.

Openssl segítségével konvertáljuk a letöltött CA tanúsítványokat PEM formátumba! Ehhez használjuk az openssl x509 parancsot! A paramétereket az alábbiak szerint válasszuk:

- Bemeneti formátum: der
- Kimeneti formátum: pem
- Bemeneti fájl: TCA3.crt / TRootCA2008.crt
- Kimeneti fájl: TCA3_pem.crt / TRootCA2008_pem.crt

Az Apache számára a befogadandó tanúsítványok láncait PEM fájlokban kell megadni.

Hozzunk létre egy fájlt auth_ca.pem néven és másoljuk bele TCA3_pem.crt és a TRootCA2008_pem.crt tartalmát egymás mögé.

SSL konfiguráció

Nyissuk meg szerkesztésre a httpd-ssl.conf állományt. és végezzük el az SSL konfigurációt!

1. Állítsuk be a SSLCACertificateFile paramétert a most létrehozott PEM tanúsítványtárra (auth_ca.pem). Az SSLCACertificatePath paraméter használatával megadhatnánk egy mappát, amiben több tanúsítvány is lehet.
2. Az SSLCARevocationPath paramétert hagyjuk kikommentezve, ebben a demóban nem fogunk foglalkozni vele. Ez a paraméter lenne alkalmas arra, hogy már letöltött CRL fájlok segítségével az Apache ellenőrizze a csatlakozó tanúsítvány érvényességét. FONTOS, hogy az Apache nem frissíti a CRL-t, egy éles környezetben ennek frissítéséről nekünk kell gondoskodnunk!
3. Vegyük ki a kommentet az SSLVerifyClient és SSLVerifyDepth paraméterek elől.
4. Indítsuk újra az Apache szervert.

Zárjuk be a böngészőt és indítsuk újra!

Látogassuk meg a https://pkilabor.crysys.hu oldalt. A böngésző ekkor kérni fogja a tanúsítvány alapú autentikációt.

FONTOS: Egy tanúsítvány alapú azonosítást követően a böngésző megjegyzi a tanúsítványt. Ha meg kívánjuk ismételni az azonosítást zárjuk be a böngészőt és látogassuk meg újra az oldalt.

Az oldal felszólít minket, hogy adjuk meg a kártya PIN kódját és válasszuk ki az autentikációs tanúsítványt.

Tanúsítványadatok kinyerése

A kliens által megadott tanúsítvánnyal való további műveletvégzésre az Apache több lehetőséget is ad számunkra.

Apache konfigurálás

Az Apache SSL beállításairól bővebben az alábbi címen lehet olvasni: https://httpd.apache.org/docs/2.4/mod/mod_ssl.html

Az SSLOptions paraméter segítségével több mindent is megtehetünk.

Nyissuk meg a httpd-ssl.conf-ot és _default_:443 virtualhost számára kapcsoljuk be az alábbi paramétereket:

- StdEnvVars: SSL-lel kapcsolatos további információk átadása a háttérrendszernek környezeti változóban.

- ExportCertData: A tanúsítvánnyal kapcsolatos további információk átadása a háttérrendszernek környezeti változóban.

(A konfigurációs állományban ezek az opciók részlegesen már benne vannak, keressük meg a megfelelő blokkot és a konfigurálást ott végezzük el! Figyeljünk arra, hogy pontosan egy aktív SSLOptions sor maradjon csak a virtualhoston belül!)

A gyakorlatban még hasznosak lehetnek az alábbi SSLOptions kapcsolók is:

- OptRenegotiate
- FakeBasicAuth

Érdemes megvizsgálni még az SSLRequire direktívát, amely a gyakorlatban további szűrési lehetőségeket ad a tanúsítványokra.

Indítsuk újra az Apache szolgáltatást!

Adatok feldolgozása PHP szkriptből

Amennyiben a konfigurációt helyesen végeztük el, mostantól az Apache szerver a háttérben futó PHP számára környezeti változóban több, a tanúsítvány alapú autentikációra vonatkozó adatot is át fog adni.

Navigáljunk a C:\xampp\htdocs mappába és írjunk PHP szkriptet pkilabor.php néven, amely kiírja a csatlakozó autentikációs tanúsítványt és a kliens Subject DN-t!

Ellenőrizze a szkript működését!

SSL tanúsítványok vizsgálata

Természetes, hogy a mérés során általunk kiadott webszerver tanúsítvány nem felel meg a nemzetközi előírásoknak és elvárásoknak. Egy hitelesítés szolgáltatótól beszerezhető webszerver tanúsítvány sokkal több adatot tartalmaz és szolgáltat, mint az, amit mi most kiállítottunk.

A pkilabor.crysys.hu tanúsítvány vizsgálata

Vizsgáljuk meg az általunk kiadott tanúsítvány alapvető tulajdonságait. Milyen fontosabb mezőket tartalmaz a tanúsítvány?

1. Nyissuk meg Chrome böngészőben az oldalunkat.
2. Navigáljunk a „Three Dots Menu → More tools → Developer Tools” ablakra.
3. Kattintsunk a Security Overview fülre.
4. Kattintsunk a tanúsítvány adatai gombra.

A részletek fülön megfigyelhetjük, hogy az általunk megadott adatok belekerültek a tanúsítványba és ezt a végfelhasználói oldalon is meg lehet tekinteni.

Az e-szigno.hu szervertanúsítvány vizsgálata

Vizsgáljuk meg az srv.e-szigno.hu webszerver tanúsítvány alapvető tulajdonságait. Milyen fontosabb mezőket tartalmaz a tanúsítvány? Miben tér el a pkilabor.crysys.hu tanúsítványától?

1. Nyissuk meg Chrome böngészőben a https://srv.e-szigno.hu címet.
2. Navigáljunk a „Three Dots Menu → More tools → Developer Tools" ablakra.
3. Kattintsunk a Security Overview fülre.
4. Kattintsunk a tanúsítvány adatai gombra.

A részletek fülön megfigyelhetjük, hogy egy, a nemzetközi elvárásoknak megfelelő Hitelesítés Szolgáltató által kibocsátott tanúsítvány miben különbözik az általunk előbb kiadott tanúsítványtól.

Aláírt PDF készítése parancssorból

Ahhoz, hogy egy elektronikusan aláírt dokumentumot létrehozzunk, az alábbiakra van szükségünk:

- Aláíratlan dokumentum
- Aláíró titkos kulcs (szoftveres, vagy hardwares eszközön tárolt)
- Aláírás létrehozó szoftver

Aláírás létrehozásra az e-Szignó termékcsaládot, valamint az Adobe Acrobat Readert fogjuk használni. Az e-Szignó termékcsaládban találhatunk egy grafikus felületű végfelhasználói szoftvert, valamint egy parancssoros klienst, melyet parancssoros, C, COM és Java interfészeken keresztül lehet integrálni az alkalmazásainkba.

Fontosabb tudnivalók

- A kliens e-Szignó alkalmazás az asztalon és a start menüben található ikonokkal indítható
- A parancssoros e-Szignó c:\eszigno3 mappában található
    - A doc mappában találhatjuk a dokumentációt
    - Az examples mappában találhatóak példák a legtipikusabb felhasználási területekre
    - A pfx mappában találhatóak a tesztekhez használható tanúsítványok
    - A bin mappában található a futtatható bináris

Készítsünk aláírt PDF-et a parancssoros e-Szignó alkalmazás segítségével. Ehhez futtassuk parancssorból a példák között található pdf_sign.bat szkriptet. Ahhoz, hogy a futtatás sikeres legyen a c:\eszigno3 mappa legyen a munkakönyvtár!

Ez a parancs létrehoz egy aláírt PDF állományt. A parancssori utasítás megadja az aláírandó állományt, az aláíráshoz használt kulcsfájlt, annak jelszavát, formátumát (pfx). Megadja a tanúsítványlánc-építéshez

használható köztes és gyökértanúsítványok mappáit, valamint az aláírás típusát, illetve az időbélyeg hozzáférés adatait arra az esetre, ha az aláírásnak tartalmaznia kellene időbélyeget.

Milyen típusú aláírásokat hoz létre a szkript? (Alap, időbélyeges, vagy archív?)

Megjegyzés: Bár a PDF aláírás szabvány önállóan is létezik PAdES néven, az aláírt PDF-ek belül CAdES típusú aláírásokat tartalmaznak.

Készítsünk egy-egy aláírt PDF-et mindhárom (alap, időbélyeges, archív) típusból!

A szoftver dokumentációját segítségül hívva módosítsuk a pdf_sign.bat állományt úgy, hogy más típusú aláírt PDF-et hozzon létre. A három állomány neve legyen:

- signed_bes.pdf
- signed_t.pdf
- signed_a.pdf

Adobe Reader konfiguráció Windows tanúsítványtár használathoz

---

Nyissuk meg valamelyik PDF-et Adobe Reader-rel.

Láthatjuk, hogy a Reader hibát jelez: "Legalább egy aláírás hibás". Ennek oka az, hogy a teszt tanúsítványainkhoz kapcsolódó teszt hitelesítés szolgáltatót az Adobe Reader (nagyon helyesen) nem ismeri el megbízhatóként.

A Windows Tanúsítványtárába a mérést megelőzően telepítésre kerültek a Microsec Teszt CA-i.

Konfiguráljuk be az Adobe Reader-t, hogy használja a Windows tanúsítványtárát!

Ehhez kövessük az alábbi lépéseket:

- Válasszuk a Szerkesztés/Beállítás menüpontot
- Válasszuk az Aláírások menüpontot
- Kattintsunk a Tovább gombra az Ellenőrzés mezőben
- Alul, az "Együttműködés a Windows rendszerrel" mezőben pipáljuk be mindkét opciót

---

Ezt követően újbóli megnyitásra az aláírás ellenőrzése sikeres lesz.

Az e-Szignóval aláírt PDF-ek vizsgálata

Nyissuk meg a három aláírt PDF-et! Az Aláírás panel segítségével hasonlítsuk össze az aláírásokat!

Milyen különbségeket tapasztalunk? Milyen új elemek jelennek meg, ahogy a dokumentumon található aláírás típusa változik?

Aláírás Adobe Readerrel

Nyissuk meg az C:\eszigno3\doc\E-Szigno_EULA.pdf állományt Adobe Readerrel! Készítsünk a dokumentumra elektronikus aláírást több különböző módon az Adobe Reader segítségével!

Ehhez használjuk az Eszközök/Tanúsítványok opciót. Készítsük el az alábbi dokumentum variációkat (Ebben a sorrendben!):

1. Csak aláírást tartalmazó dokumentum: eula_bes.pdf
2. Csak időbélyeget tartalmazó dokumentum: eula_ts.pdf
3. Csak aláírást tartalmazó dokumentum (és adjuk meg az időbélyeg adatokat amikor kéri): eula_t.pdf
4. Időbélyeg, majd aláírás elhelyezése: eula_ts_and_t.pdf
5. Aláírás, majd időbélyeg elhelyezése: eula_t_and_ts.pdf

Időbélyegzéshez az alábbi kiszolgálót állítsuk be:

- Kiszolgáló URL: https://bteszt.e-szigno.hu/tsa
- Felhasználónév: teszt
- Jelszó: teszt

Hasonlítsuk össze az így keletkezett dokumentumokat az e-Szignóval készültekkel!

Ha eddig még nem tettük volna, most olvassuk el a parancssoros e-Szignó dokumentációjában (eszigno3_ref.html) a pdf_sign parancs leírását.

Próbáljunk megfeleltetést találni a releváns e-Szignó paraméterek (cades_type = bes,t,a) és Adobe Reader két funkciójával kombinációival elérhető eredmények között.

Láthatjuk, hogy bizonyos Adobe Reader-rel készített állományok egy- az-egyben megfeleltethetőek az e-Szignó által készítetteknek. Vannak azonban olyan fájlok, amelyekhez nem található e-Szignós állomány.

Mi az a funkció, amit az e-Szignó parancsok nem láttak el? Melyik e-Szignó parancs segítségével lehetséges a hiányzó funkciót betölteni?