Electronic Signatures - Tasks

Iulian Aciobanitei, Phd

Bucharest

2024

Schedule

Useful Links

1. Alice VM
2. Virtual Box
3. OpenSSL
4. XML Signer
5. PKCS#7 Signer

Deliverables

1. Solve as many checkboxes as you can
2. Document everything with screenshots in a doc/docx file
3. Convert it into a PDF
4. Sign the obtained PDF
5. Rename it to HW_ES_2024_'FamilyName'_'GivenName'.pdf (ex: HW_ES_2024_Popa_Ana.pdf)
6. Send it to teme.aciobanitei.iulian@gmail.com
   Email Subject: HW_ES_2024_'FamilyName'_'GivenName'

Environment

1. Download Alice VM (user: kali, passwd: kali)
2. Use Virtual Box to start it. Set Host-Only then reboot.
3. Start EJBCA Application
   - > cd Desktop/ejbca
   - > sudo docker-compose up
4. Get SuperAdmin Certificate and install it in windows store. (password 1234)
5. Access EJBCA's public interface: `http://<<vm_ip>>:8080/ejbca`
6. Access EJBCA's administrative interface: `https://<<vm_ip>>:8443/ejbca/adminweb`

Certificates

1. Download this certificate
2. Download all the certificates in the certification path
3. Download the CRL manually
4. Verify the certificate using certutil and CRL
5. Verify the certificate using certutil and OCSP
6. Verify the certificate using openssl*
7. View certificate using an ASN.1 viewer (openssl cmd or lapo.it)
8. Extract the OID of an atribute from Subject extension, at will.

*Example of openssl cmd(admin): openssl ocsp -issuer ca_cert.cer -cert cert.cer -no_nonce -url "ocsp url" -CAfile root.cer

Get a signing certificate

1. From EJBCA's admin interface, create a new end entity of type User_Sign or TLS_Client
2. From EJBCA's public interface, actually generate the certificate and obtain the .p12 file.
3. Import the generated certificate into the Windows Store.

PDF Signatures

1. Sign a document
2. Validate a document
3. Search for a free timestamp server. Hint: github
4. Apply a timestamp to the document
5. Validate the document and check the signature format this time

XML Signatures

1. Sign an XML document
2. Validate the signed document
3. Validate an altered version of the document

PKCS#7 Signatures

1. Use a media document - a picture, a text file, or other file type.
2. Obtain an attached signature
3. Validate it
4. Obtain a detached signature
5. Validate it
6. Check the obtained file with ASN.1 viewer.

Microsoft Word Signatures

1. Sign a document using Microsoft Word.

Mozilla vs Microsoft Store

1. Prove that we are using different stores to validate the certificate of the web server.
2. We can use https://testssl.certsign.ro/
3. Hint: Should use different browsers (Mozilla+Chrome, for example)

Certificate Requests

1. Generate a certificate request using openssl.
2. Use the generated CSR to issue a new certificate, from EJBCA
3. Use openssl to create a pkcs12 file from certificate and private key
4. Import the certificate to Windows store
5. Try to sign a PDF file

Useful commands:

- openssl req -newkey rsa:2048 -keyout private.key -out openssl.csr
- openssl pkcs12 -export -in encryption.crt -inkey enc_private.key -out encryption.p12

CSC Protocol - 'Hello world' level

Obtain one of the cloud certificates of the user with phone number +40767367791, using the CSC protocol.
Helping details:

1. CSC API is exposed here.
2. CSC API documentation can be found here
3. You need to call /credentials/list and credentials/info

Bonus - Python Certificate Requests

1. Write a python script to create a CSR
2. Use the generated CSR to issue a new certificate, from EJBCA
3. Use openssl to create a pkcs12 file from certificate and private key
4. Import the certificate to Windows store
5. Try to sign a PDF file

Bonus - Coding

- Write a python script that can create a PKCS#7 or an XML signature.
  - Note: Usually, you might face more challenges on XML sigs.
- You can use P7s Viewer or XML Signer Viewer applications in order to validate your implementation
- You may use .p12 file or cert+key files for signing