

X.509 verifier, RESTful Web Servisi

[RS] Uputstvo za upotrebu

"X.509 verifier" predstavlja RESTful Web servise koji mogu poslužiti za proveru validnosti X.509 sertifikata i potpisanih PDF fajlova kao i proveru usaglašenosti sadržaja sertifikata sa regulativom Republike Srbije.

"X.509 verifier" RESTful Web servise čine 2 REST API-ja koji su locirani na hostu:

<http://signatureverifier.d-logic.com>

I čije su putanje:

1. [/x509-verifier.php](#)
2. [/pdf-sgn-verifier.php](#)

REST API: x509-verifier

API verzija: 1.0

Ovom API-ju se šalje PEM fajl čiji sadržaj mora biti X.509 sertifikat verzije 3. Nakon provere sertifikata prosleđenog serveru, API vraća rezultat provere u JSON enkodiranom stringu.

Zahtev koji se šalje HTTP serveru

Host + Path: <http://signatureverifier.d-logic.com/x509-verifier.php>

Method: POST

Headers (mandatory):

Content-Type: multipart/form-data; boundary=RANDOM_STRING_BOUNDARY

Body:

--RANDOM_STRING_BOUNDARY

Content-Disposition: form-data; name="file"; filename="file_name.pem"

Content-Type: application/octet-stream

[FILE_BINARY_DATA]

--RANDOM_STRING_BOUNDARY

Content-Disposition: form-data; name="query"

[JSON_ENCODED_PARAMETERS]

--RANDOM_STRING_BOUNDARY--

{END}

Opis zahteva koji se šalje HTTP serveru

RANDOM_STRING_BOUNDARY je string koji treba prilikom svakog novog zahteva da ima drugačiju i po mogućstvu jedinstvenu vrednost. Dobar način za dobijanje ovog stringa u JavaScript-u bio bi:

```
var boundary = Math.random().toString().substr(2);
```

[FILE_BINARY_DATA] je binarni sadržaj odabranog 'file_name.pem' fajla.

[JSON_ENCODED_PARAMETERS] su JSON enkodirani parametri koji moraju da budu u formatu:

```
{
  "operation": "verify",
  "user_id": 123,
  "security_token": ""
}
```

i najbolje je da ovaj JSON enkodirani string bude bez tzv. "whitespace" karaktera odn. da u JavaScript-u bude formiran na sledeći način:

```
var params, json;
params= { operation: "verify", user_id: 123, security_token: "" };
json = JSON.stringify(params);
```

Ovde su parametri:

"operation": "verify" - za sada je operacija "verify" jedino podržana.

"user_id": 123 - numerički parametar, iz skupa prirodnih brojeva i predstavlja identifikacioni broj korisnika prijavljenog na sistem (ne proverava se u verziji servisa 1.0 ali je obavezan jer je predviđen za buduću upotrebu).

"security_token": "" - predviđen je da bude string koji sadrži parove heksadecimalnih cifara bez tzv. Delimitera. Ne proverava se u verziji servisa 1.0 pa može ostati prazan string (""). Obavezan je, jer je predviđen za buduću upotrebu.

U svakom slučaju, kada se koristi JavaScript, nije neophodno direkno baratati sa samim Content-om već je preporuka koristiti FormData klasu, kao u primeru koji možete pronaći na git repozitorijumu:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/signature_verifier_jc_example.git

Postoje i primeri za PHP REST klijente ovih API-ja uz korišćenje cURL podrške u PHP-u:

https://www.d-logic.net/code/digital_signature_sdk/php_example.git

Odziv HTTP servera

Nakon provere X.509 sertifikata, server će vratiti JSON enkodirani string koji (u verziji API-ja 1.0) sadrži 2 argumenta:

```
{"status": "STATUS_STRING", "msg": "MESSAGE_STRING" }
```

Ukoliko zahtev nije pravilno formatiran, server vraća:

```
HTTP/1.1 200 OK
```

```
...
```

```
Content-Type: application/json
```

```
{"status": "Error: wrong POST parameters.", "msg": ""}
```

Ako je provera **uspešna**, STATUS_STRING mora biti:

```
"OK"
```

dok će MESSAGE_STRING sadržati validno formatirani zapis, koji ima html tagove za formatiranje i boju fonta, kao i html tagove za novi red, tako da se ova poruka može postaviti u bilo koji html kontejner (npr. <div>) uz eventualno definisanje jedino osnovnog tipa i veličine fonta.

Bilo koji odziv čiji je STATUS_STRING različit od "OK" računa se kao provera čiji je rezultat **neuspešan** i u tom slučaju, ukoliko je STATUS_STRING različit i od "Error: wrong POST parameters.", MESSAGE_STRING će sadržati detalje provere koje takođe treba prikazati.

REST API: pdf-sgn-verifier

API verzija: 1.0

Ovom API-ju se šalje PDF fajl čiji sadržaj mora biti potpisani PDF dokument. Formati potpisa mogu biti "PKCS#7 - Detached" ili "CAAdES Equivalent". Nakon provere potpisanog fajla prosleđenog serveru, API vraća rezultat provere u JSON enkodiranom stringu.

Zahtev koji se šalje HTTP serveru

Host + Path: <http://signatureverifier.d-logic.com/pdf-sgn-verifier.php>

Method: POST

Headers (mandatory):

Content-Type: multipart/form-data; boundary=RANDOM_STRING_BOUNDARY

Body:

--RANDOM_STRING_BOUNDARY

Content-Disposition: form-data; name="file"; filename="file_name.pdf"

Content-Type: application/pdf

[FILE_BINARY_DATA]

--RANDOM_STRING_BOUNDARY

Content-Disposition: form-data; name="query"

[JSON_ENCODED_PARAMETERS]

--RANDOM_STRING_BOUNDARY--

{END}

Opis zahteva koji se šalje HTTP serveru

RANDOM_STRING_BOUNDARY je string koji treba prilikom svakog novog zahteva da ima drugačiju i po mogućstvu jedinstvenu vrednost. Dobar način za dobijanje ovog stringa u JavaScript-u bio bi:

```
var boundary = Math.random().toString().substr(2);
```

[FILE_BINARY_DATA] je binarni sadržaj odabranog 'file_name.pdf' fajla.

[JSON_ENCODED_PARAMETERS] su JSON enkodirani parametri koji moraju da budu u formatu:{

```
  "operation": "verify",
  "user_id": 123,
  "security_token": ""
}
```

i najbolje je da ovaj JSON enkodirani string bude bez tzv. "whitespace" karaktera odn. da u JavaScript-u bude formiran na sledeći način:

```
var params, json;
params= { operation: "verify", user_id: 123, security_token: "" };
json = JSON.stringify(params);
```

Ovde su parametri:

"operation": "verify" - za sada je operacija "verify" jedino podržana.

"user_id": 123 - numerički parametar, iz skupa prirodnih brojeva i predstavlja identifikacioni broj korisnika prijavljenog na sistem (ne proverava se u verziji servisa 1.0 ali je obavezan jer je predviđen za buduću upotrebu).

"security_token": "" - predviđen je da bude string koji sadrži parove heksadecimalnih cifara bez tzv. Delimitera. Ne proverava se u verziji servisa 1.0 pa može ostati prazan string (""). Obavezan je, jer je predviđen za buduću upotrebu.

U svakom slučaju, kada se koristi JavaScript, nije neophodno direkno baratati sa samim Content-om već je preporuka koristiti FormData klasu, kao u primeru koji možete pronaći na git repozitorijumu:

https://www.d-logic.net/code/nfc-rfid-reader-sdk/signature_verifier_jc_example.git

Postoje i primeri za PHP REST klijente ovih API-ja uz korišćenje cURL podrške u PHP-u:

https://www.d-logic.net/code/digital_signature_sdk/php_example.git

Odziv HTTP servera

Nakon provere PDF fajla i potpisa koji sadrži, server će vratiti JSON enkodirani string koji (u verziji API-ja 1.0) sadrži 2 argumenta:

```
{"status": "STATUS_STRING", "msg": "MESSAGE_STRING"}
```

Ukoliko zahtev nije pravilno formatiran, server vraća:

```
HTTP/1.1 200 OK
```

```
...
```

```
Content-Type: application/json
```

```
{"status": "Error: wrong POST parameters.", "msg": ""}
```

Ako je provera **uspešna**, STATUS_STRING mora biti:

```
"PDF Signature is VALID"
```

dok će MESSAGE_STRING sadržati validno formatirani zapis, koji ima html tagove za formatiranje i boju fonta, kao i html tagove za novi red, tako da se ova poruka može postaviti u bilo koji html kontejner (npr. <div>) uz eventualno definisanje jedino osnovnog tipa i veličine fonta.

Za razliku od x509-verifier API-ja postoje odzivi čiji je STATUS_STRING različit od "PDF Signature is VALID", koji se računaju kao provere čiji je rezultat **neuspešan** ali **MESSAGE_STRING ne sadržati detalje provere**. To su slučajevi kada je:

```
STATUS_STRING = "Error: PDF was changed after signing!"  
STATUS_STRING = "Error: Wrong PDF format (while searching signature data)"  
STATUS_STRING = "Info: PDF file doesn't contain digital signature"  
STATUS_STRING = "Error: Wrong PKCS#7 format (missing \"to be signed\" data)"
```

U slučaju da je

```
STATUS_STRING = "Digital signature validation FAILED"
```

znači da je provera PDF fajla i potpisa koji sadrži **urađena do kraja** ali da je njen **rezultat neuspešan**. U ovom slučaju MESSAGE_STRING **uvek sadrži detalje provere koje treba prikazati**. I u ovom slučaju MESSAGE_STRING ima html tagove za formatiranje i boju fonta, kao i html tagove za novi red, tako da se ova poruka može postaviti u bilo koji html kontejner (npr. <div>) uz eventualno definisanje jedino osnovnog tipa i veličine fonta.

Dodatak: Eksportovani fajlovi za "Restlet Client", ekstenziju "Google Chrome" pretraživača Interneta

Prateći deo ovog uputstva predstavljaju i JSON fajlovi koji sadrže profile za "Google Chrome" ekstenziju "Restlet Client". Ovi fajlovi su **x509-verifier.json** i **pdf-sgn-verifier.json** i sadrže profile za REST API-je **x509-verifier** i **pdf-sgn-verifier** respektivno. Mogu se koristiti nakon instaliranja "Restlet Client" "Google Chrome" ekstenzije.

Ovaj "Restlet Client" ima grešku uvek kada se izabere "Request" koji sadrži podešen Content-Type: multipart/form-data i "Form" u Body-u. Greška se manifestuje tako što uvek po izboru sačuvanog zahteva, u "Body" delu definicije tog zahteva, za item "file" promeni "Item Type" sa podešenog "File" na "Text". Sve što treba uraditi je da se za item "file" vrati "Item Type" na "File" i odabere željeni fajl.

Dodatak: Dnevnik promena

datum	opis	revizija	odnosi se na API verzije x509-verifier / pdf-sgn-verifier
2019-03-12	Predmet ove revizije je dokument preveden na engleski jezik.	1.3	1.0 / 1.0
2019-01-25	Kozmetičke promene. Od ove revizije postoji i prevod na engleski jezik.	1.2	1.0 / 1.0
2019-01-19	<ul style="list-style-type: none"> Dodate su reference na PHP cURL primere u okviru odeljaka "Opis zahteva koji se šalje HTTP serveru". Promenjen naslov 'Appendix: "Restlet Client" - Google Chrome browser extension exported files' zbog nejasnoća u vezi kopiranja JSON listinga iz dokumenta i "pejstovanja" u JSON fajlove. JSON fajlovi su prateći deo ovog dokumenta i moraju se nalaziti u istom folderu. 	1.1	1.0 / 1.0
2019-01-10	Prvo izdanje	1.0	1.0 / 1.0