

eHerkenning rapport Enable-u

Enable-U 2Secure

Auteur: Kevin Rijs

Plaats en Datum: Amsterdam, 03-10-2016

Versie: 1.0

1. Inhoudsopgave

1.	Inhoudsopgave	2
2.	eHerkenning aansluiting Enable-u	3
2.1	Genereren en signen metadata Dienstverlener	3
2.2	Versturen en signen SAML/Authn Request	3
2.3	Ontvangen en valideren SAML/Authn Response	4

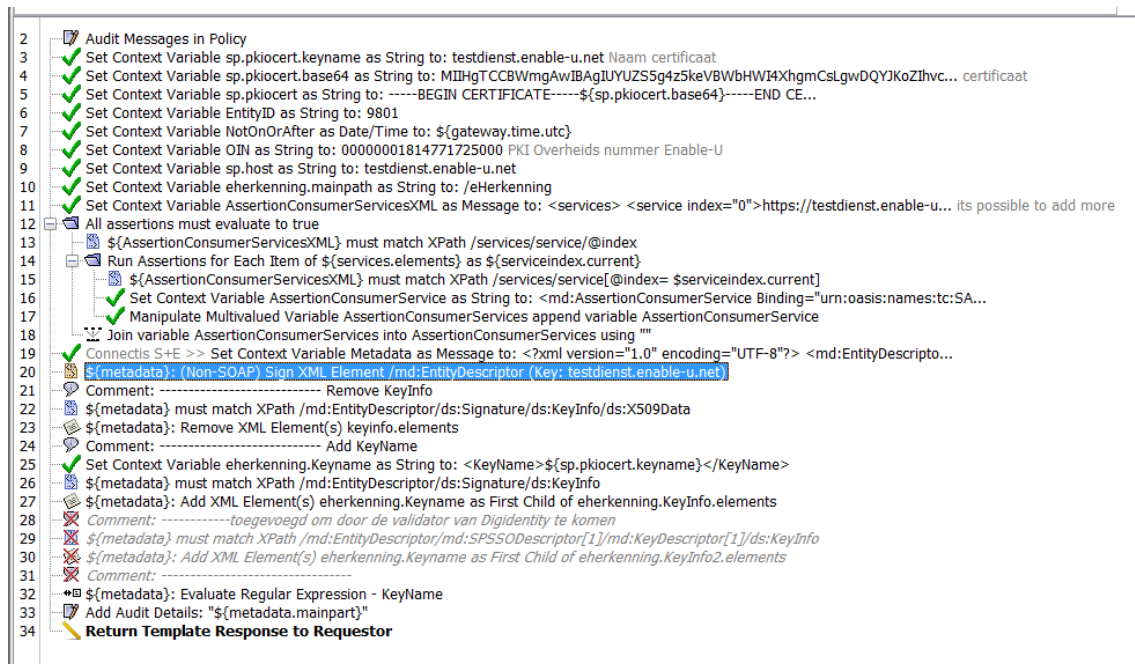
2. eHerkenning aansluiting Enable-u

De eHerkenning aansluiting van Enable-u wordt gerealiseerd door Enable-u2Secure (powered by CA API Management Gateway). De belangrijkste stappen in het authenticatie proces van de dienstverlener worden gerealiseerd in de zogehete 'polities' van Enable-u 2Secure waar verschillende stappen worden afgedwongen en gevalideerd. De stappen in de eHerkenning policy bestaan uit individuele regels die van boven naar onder worden gelezen.

Hieronder worden kort de belangrijkste stappen van het Eherkenning inlog proces beschreven.

2.1 Genereren en signen metadata Dienstverlener

Hieronder is een screenshot van de policy aanwezig die laat zien hoe de XML van de metadata van de Dienstverlener wordt opgebouwd (regel 2-19) op het moment dat deze wordt aangeroepen. Nadat deze is gegenereerd wordt deze gesigned met het private key van de klant (regel 20) en de signature in de metadata geplaatst (regel 21-32). Daarna wordt de volledig gegenereerde metadata aan de requestor getoond (regel 34).



2.2 Versturen en signen SAML/Authn Request

Hieronder is een screenshot van de policy aanwezig die laat zien hoe een Authn/SAML request wordt opgebouwd. Als eerst wordt de relaystate en het samlrequest message gemaakt (regel 63-70). Hierna wordt de samlRequest gesigned (regel 71), waarna de signature in het saml eHerkenning rapport Enable-u

request wordt geplaatst (regel 73-82). Als laatste wordt de gebruiker geredirect naar de Herkennings Makelaar (regel 83-87).

```

58  [+] Include Policy Fragment: eHerkenning - Bouwblok - Zoek HM metadata op
59  [!] At least one assertion must evaluate to true
63  [i] Comment: ----- Set Relay state -----
64  [x] Set Context Variable relaystate as String to: ${request.url.path}${request.url.query}
65  [x] Set Context Variable relaystate as String to: ${routing.url}
66  [i] Add Audit Details: "relayState ${Relaystate}"
67  [i] Base64 Encode ${RelayState} into ${encodedRelayState}
68  [i] Comment: ----- Build, sign and encode SAML Request message -----
69  [x] Set Context Variable SAMLRequest as Message to: <?xml version="1.0" encoding="UTF-8"?><samlp:AuthnRequest...
70  [x] Set Context Variable SAMLRequest as Message to: <?xml version="1.0" encoding="UTF-8"?><samlp:AuthnRequest... Werk voor Connecti
71  [x] ${samlRequest}: (Non-SOAP) Sign XML Element /samlp:AuthnRequest (Key: testdienst.enable-u.net)
72  [x] Include Policy Fragment: eHerkenning - Bouwblok - Move issuer above signature
73  [x] ${samlRequest}: Select Single Element /samlp:AuthnRequest/ds:Signature/ds:KeyInfo/ds:X509Data to ${X509DataElement}
74  [x] ${samlRequest}: Remove XML Element(s) X509DataElement
75  [x] Look Up Trusted Certificate by Name: ${sp.host}
76  [x] Set Context Variable KeyName as String to: ${cert.subject.cn}
77  [x] ### Set Context Variable KeyName as String to: testdienst.enable-u.net
78  [x] Set Context Variable eHerkenning.Keyname as String to: <KeyName>${KeyName}</KeyName>
79  [x] ${samlRequest} must match XPath /samlp:AuthnRequest/ds:Signature/ds:KeyInfo
80  [x] ${samlRequest}: Add XML Element(s) eHerkenning.Keyname as First Child of eHerkenning.KeyInfo.elements
81  [x] Base64 Encode ${samlRequest} into ${encodedSamlRequest}
82  [i] Add Audit Details: "saml request: ${samlRequest.mainpart}"
83  [i] Comment: ----- Redirect to eHerkenning IDP -----
84  [x] Set Context Variable nextRedirectState as String to: samlArt
85  [x] Set Context Variable responsebody as Message to: <?xml version="1.0" encoding="UTF-8"?><!DOCTYPE html PUBL...
86  [x] Set Context Variable responsestatus as String to: 200
87  [x] Return Template Response to Requestor
  
```

2.3 Ontvangen en valideren SAML/Authn Response

Hieronder is een screenshot van de policy aanwezig die laat zien hoe een ontvangen Authn/Saml response wordt gevalideerd en gelezen. Als eerst wordt er gezocht naar de signature van de herkenningsmakelaar (regel 129-132), waarna deze controleerd wordt tegen het desbetreffende certificaat van de herkennings makelaar (regel 133). Daarna wordt het saml response gevalideerd op structuur (regel 136) en wordt er gekeken of het betrouwbaarheidsniveau overeenkomt met het benodigde niveau (regel 137-141). Is dit het geval dan worden alle meegestuurde attributen van de gebruiker uitgelezen (regel 142-165) en later aangeboden aan de backend applicatie.

```

125  [!] All assertions must evaluate to true Controleer SAMLResponse, vraag SAML token op en controleert SAML token
126  [x] ${samlResponse} must match XPath /samlp:Response/samlp:Status/samlp:StatusCode[@Value=urn:oasis:names:tc:SAML:2.0:status:Success]
127  [x] Include Policy Fragment: eHerkenning - Bouwblok - Zoek HM metadata op
128  [x] Include Policy Fragment: eHerkenning - Bouwblok - Zoek SP metadata op
129  [i] Comment: ----- Check SAML assertion integrity and validity period-----
130  [x] ${samlResponse} must match XPath count(/samlp:Response/saml:Assertion)=1
131  [x] ${samlResponse} must match XPath /samlp:Response/saml:Assertion
132  [x] ${samlResponse} must match XPath count(/samlp:Response/ds:Signature)=1 Assertion voor het controleren van de handtekening (Toegevoegd na testen)
133  [x] ${samlResponse}: (Non-SOAP) Verify XML Element /samlp:Response/ds:Signature Assertion voor het controleren van de handtekening (niet saml testen)
134  [i] Add Audit Details: "signature check succesful"
135  [x] Set Context Variable assertion as Message to: ${assertion.element}
136  [x] ${assertion}: (Non-SOAP) Validate SAML Token v2 Authentication Statement
137  [i] Comment: ----- Check SAML assertion zekerheidsniveau -----
138  [x] ${samlResponse} must match XPath /samlp:Response/saml:Assertion/saml:AuthnStatement/saml:AuthnContext/saml:AuthnContextClassRef
139  [x] Set Context Variable AuthnContextClassRef as String to: ${AuthnContextClassRef.result}
140  [i] Add Audit Details: "AuthnContextClassRef: ${AuthnContextClassRef}"
141  [x] Map AuthnContextClassRef Map Value >> Map AuthnContextClassRef to betrouwbaarheidsniveau
142  [i] Comment: ----- Check SAML assertion -----
143  [x] Include Policy Fragment: eHerkenning - Bouwblok - Valideer zekerheidsniveaus
144  [x] Set Context Variable samlAttributes as Message to: <attributes></attributes>
145  [x] ${samlAttributes}: Select Single Element /attributes to ${attributesXML}
146  [i] Comment: ----- Pseudoniem -----
147  [x] ${samlResponse} must match XPath /samlp:Response/saml:Assertion/saml:Subject/saml:NameID
148  [x] Set Context Variable eHerkenningAttribute as String to: Pseudoniem/${Pseudoniem.result}
149  [i] Add Audit Details: "Attribute: ${eHerkenningAttribute}"
150  [x] Set Context Variable samlAttribute as Message to: <attribute Name="Pseudo">${pseudoniem.result}</attribute>
151  [x] ${samlAttribute}: Select Single Element /attribute to ${samlAttribute}
152  [x] ${samlAttributes}: Add XML Element(s) samlAttribute as First Child of attributesXML
153  [i] Comment: ----- Attributen -----
154  [x] ${samlResponse} must match XPath /samlp:Response/saml:Assertion/saml:AttributeStatement/saml:Attribute/@Name
155  [x] Run Assertions for Each Item of ${XMLAttribute.elements} as ${Attribute.current}
156  [x] ${samlAttributes} must match XPath /attributes
165  [i] Add Audit Details: "Attribute: ${samlAttributes.elements}"
  
```