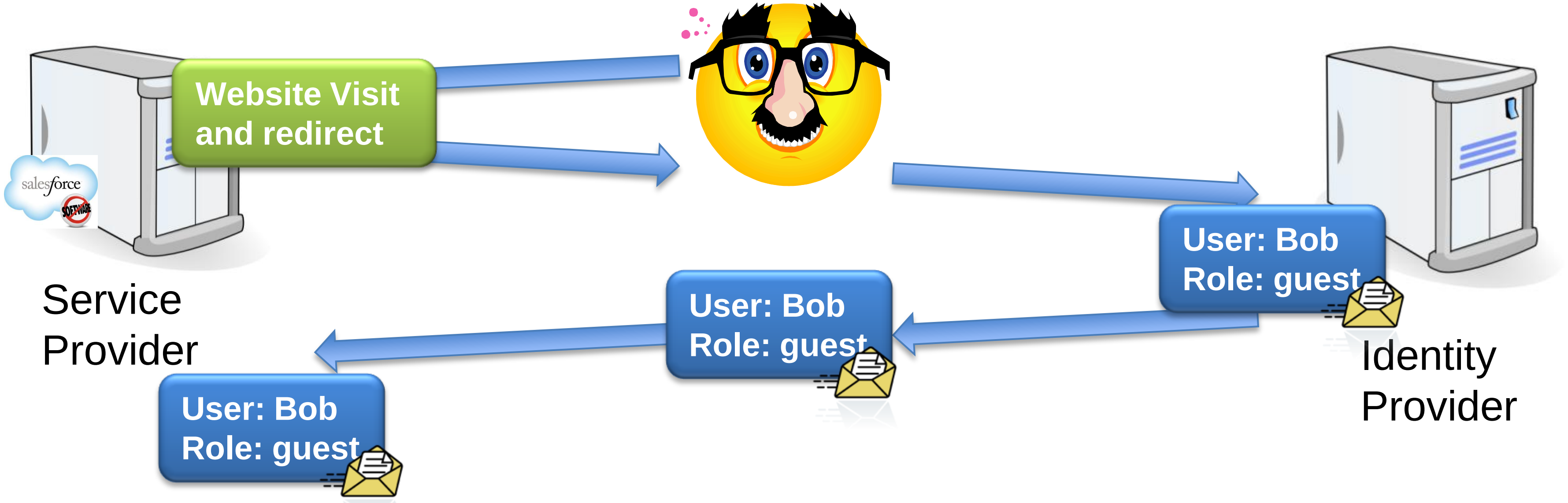
On Breaking SAML: Be Whoever You Want to Be

Juraj Somorovsky[1], Andreas Mayer[2], Jörg Schwenk[1], Marco Kampmann[1], and Meiko Jensen[1]

[1]Horst-Görtz Institute for IT-Security, Ruhr-University Bochum
[2]Adolf Würth GmbH & Co. KG

Motivation – Single Sign-On

- Too many identities / passwords
- Solution: Single Sign-On



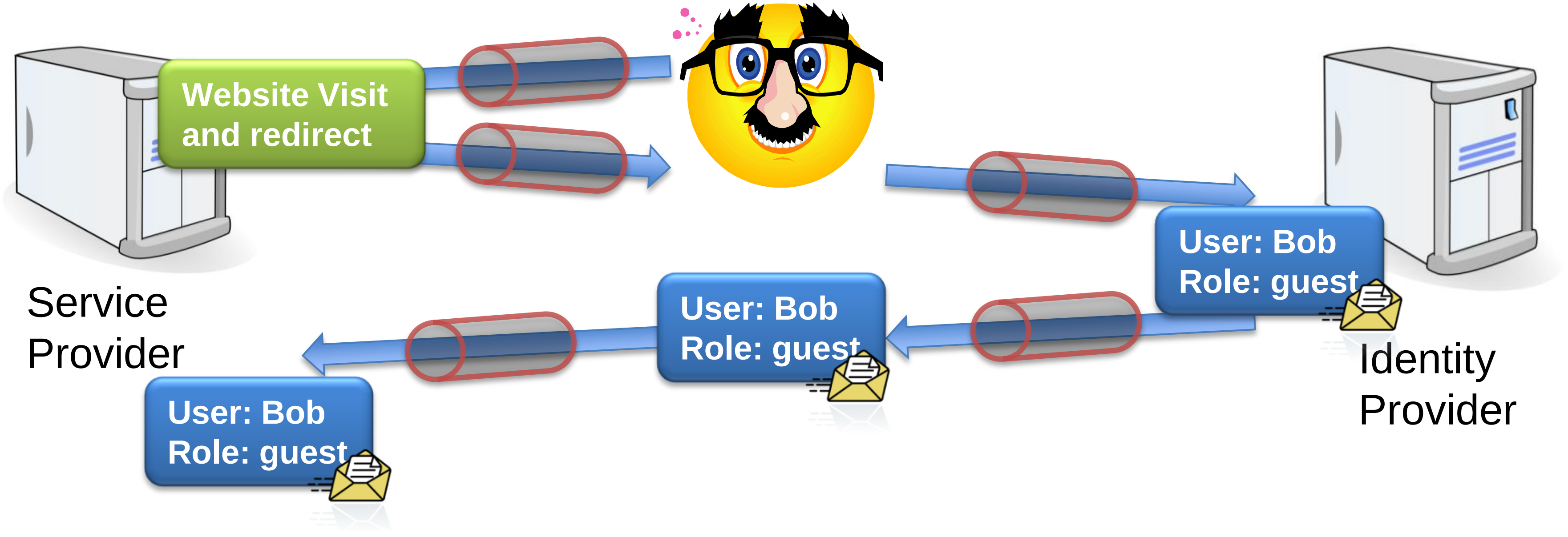
- Advantages: one password for users, no password management for Service Providers

Motivation – Single Sign-On

- OpenID
- OAuth
- **Security Assertion Markup Language (SAML)**
  - OASIS
  - Web Services or browser-based Single Sign-On
  - Authentication Statements stored in *Assertions*

Motivation – Single Sign-On

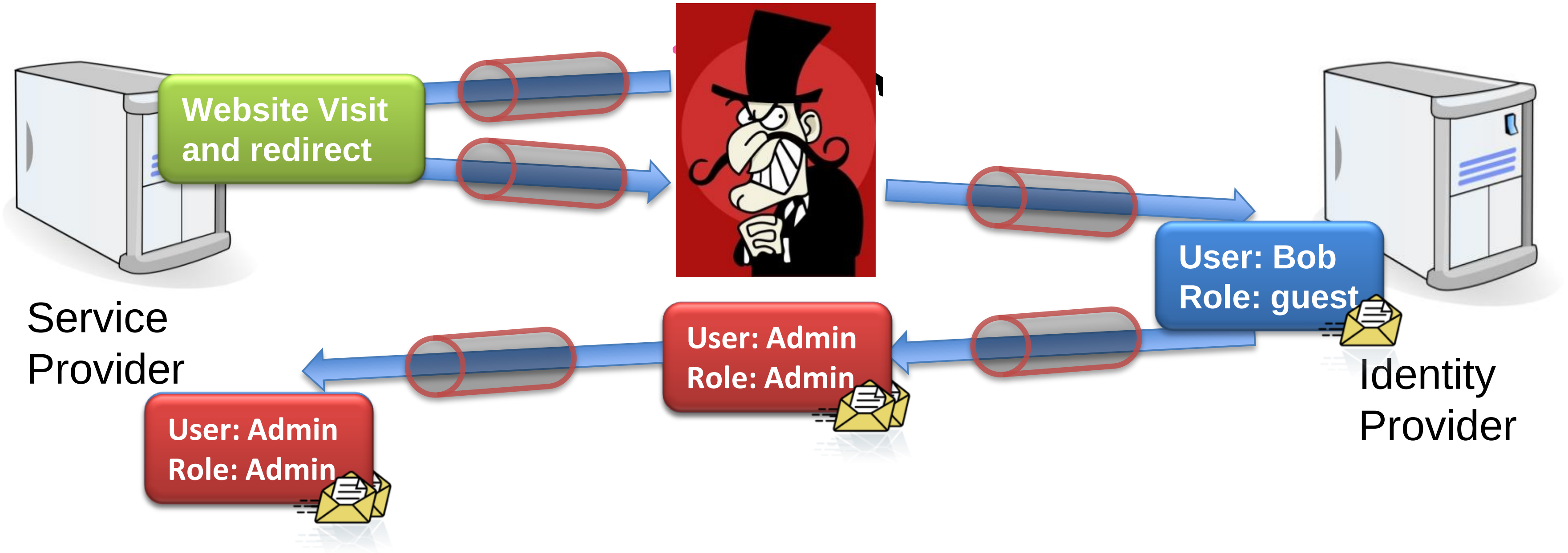
- How do we secure the messages?
- Does SSL / TLS help?



- Messages secured only during transport!

Motivation – Single Sign-On

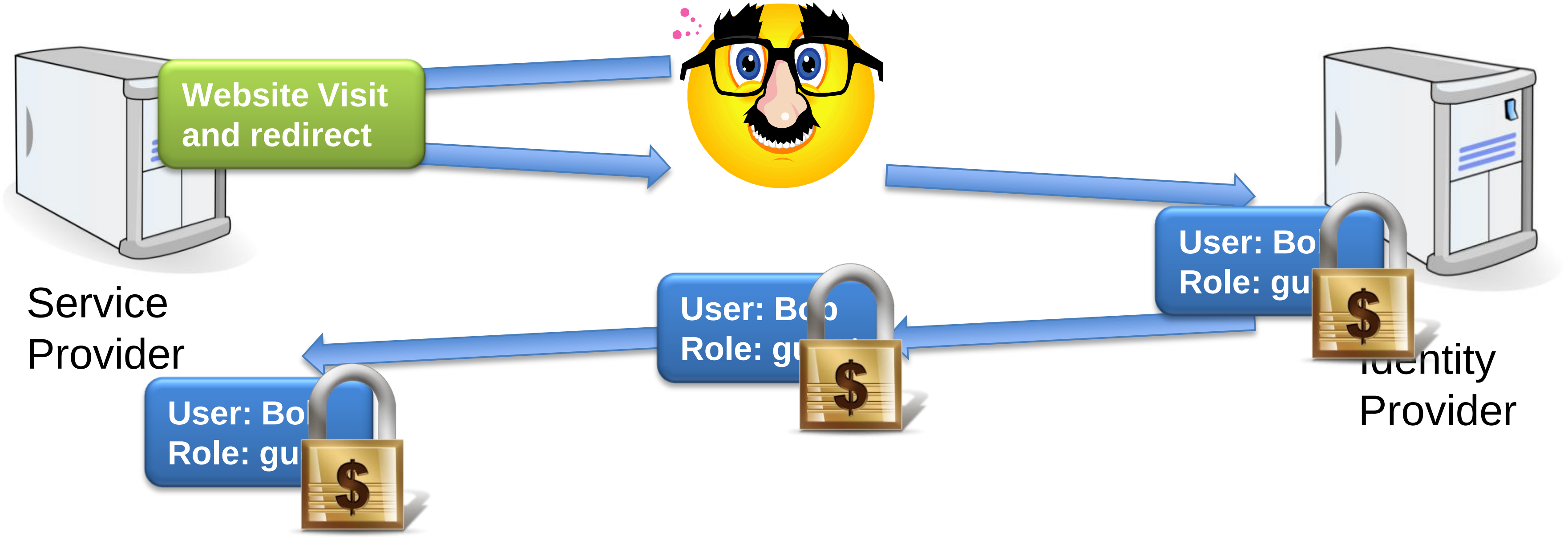
- Does SSL / TLS help?



- Need for message level security!

Motivation – Single Sign-On

- Message level security?



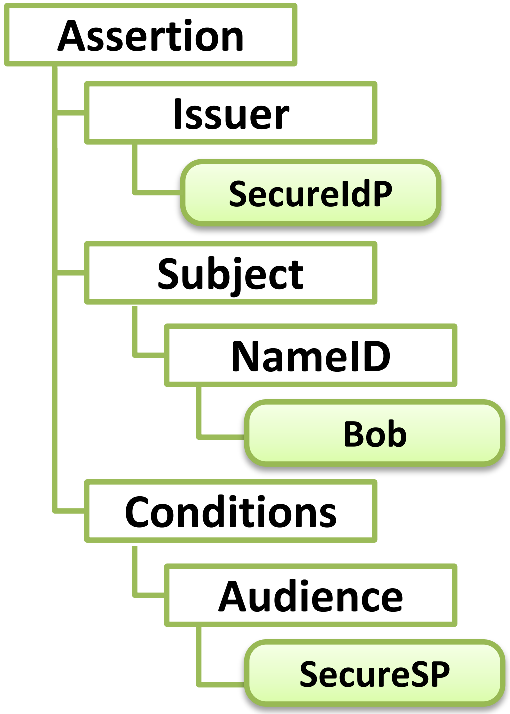
- Realized using XML Signatures
- Are we secure?

Overview

1. Securing SAML with XML Signature
2. XML Signature Wrapping Attacks
3. Practical Evaluation
4. Penetration Test Library
5. Countermeasures
6. Conclusion

SAML Assertion

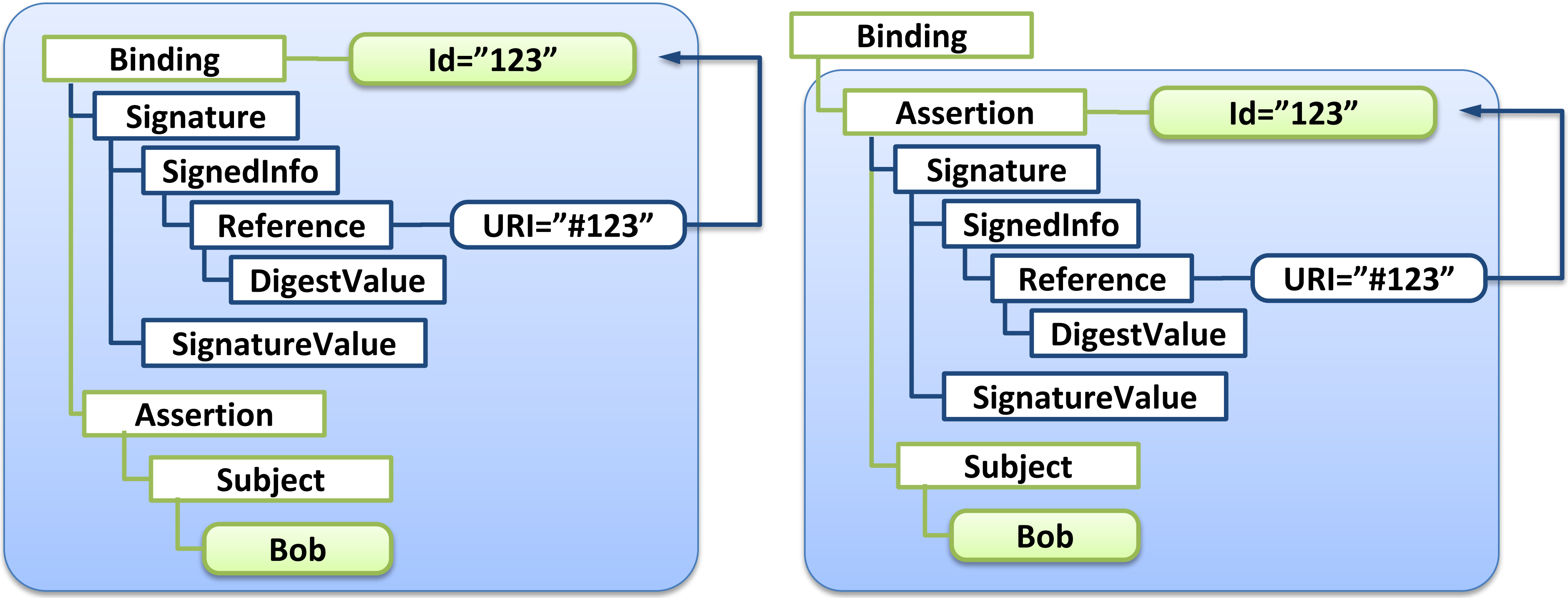
```
<saml:Assertion ID="123">
  <saml:Issuer>www.SecureIdP.com</saml:Issuer>
  <saml:Subject>
    <saml:NameID>Bob@SecureIdP.com</saml:NameID>
  </saml:Subject>
  <saml:Conditions
     NotBefore="2011-08-08T14:42:00Z"
     NotOnOrAfter="2011-08-08T14:47:00Z">
    <saml:AudienceRestriction>
      <saml:Audience>
       www.SecureSP.com</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>
</saml:Assertion>
```



Assertion
Issuer
SecureIdP
Subject
NameID
Bob
Conditions
Audience
SecureSP

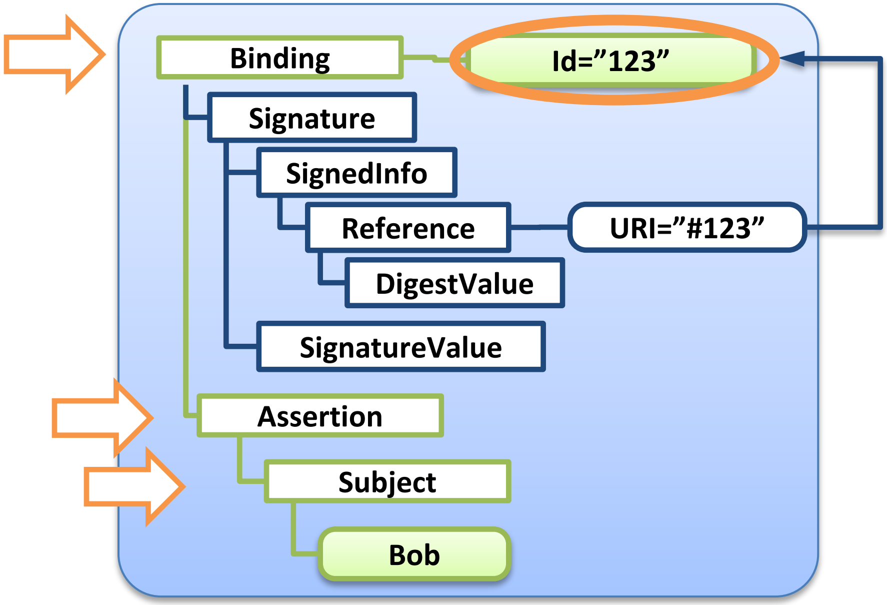
Securing SAML with XML Signature

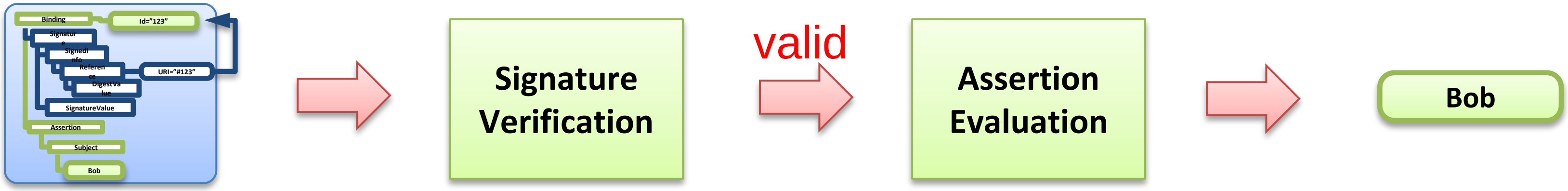
- Two typical usages



Securing SAML with XML Signature

- Naive (typical) processing:
  1. Signature validation: **Id-based**
  2. Assertion evaluation: **/Binding/Assertion/Subject**

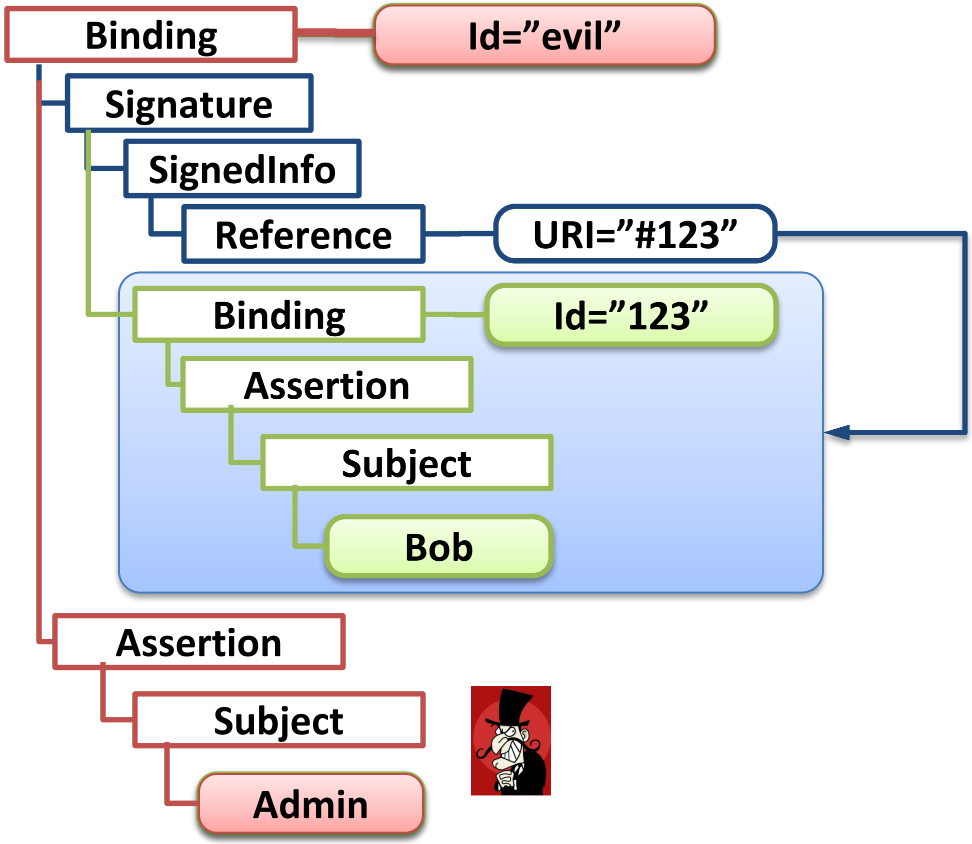




Overview

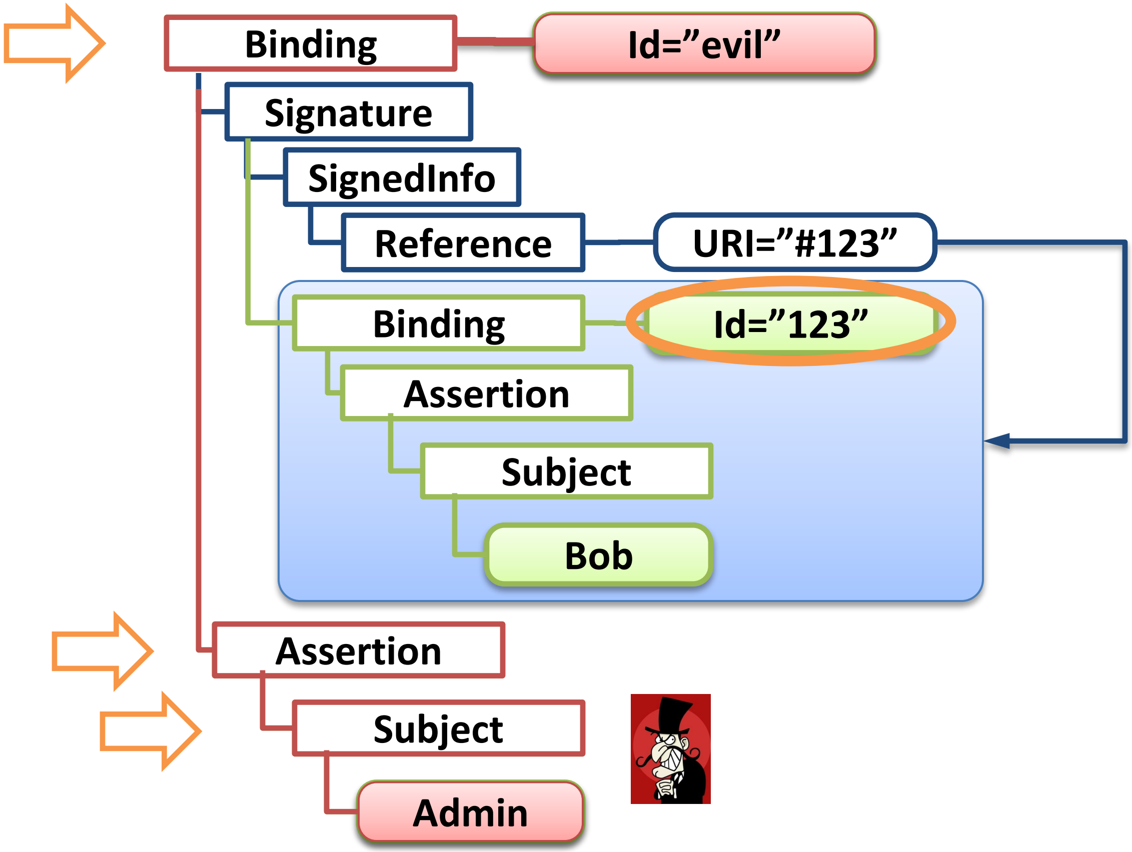
1. Securing SAML with XML Signature
2. XML Signature Wrapping Attacks
3. Practical Evaluation
4. Penetration Test Library
5. Countermeasures
6. Conclusion

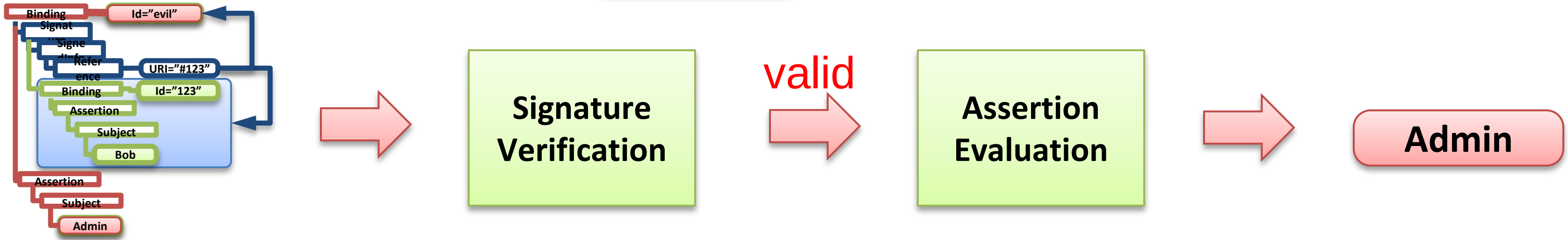
XML Signature Wrapping Attack on SAML



1. Place the original Assertion including its Binding element into another element
2. Change the Id of the original element
3. The Reference now points to the original element: signature is valid
4. Insert a new Assertion

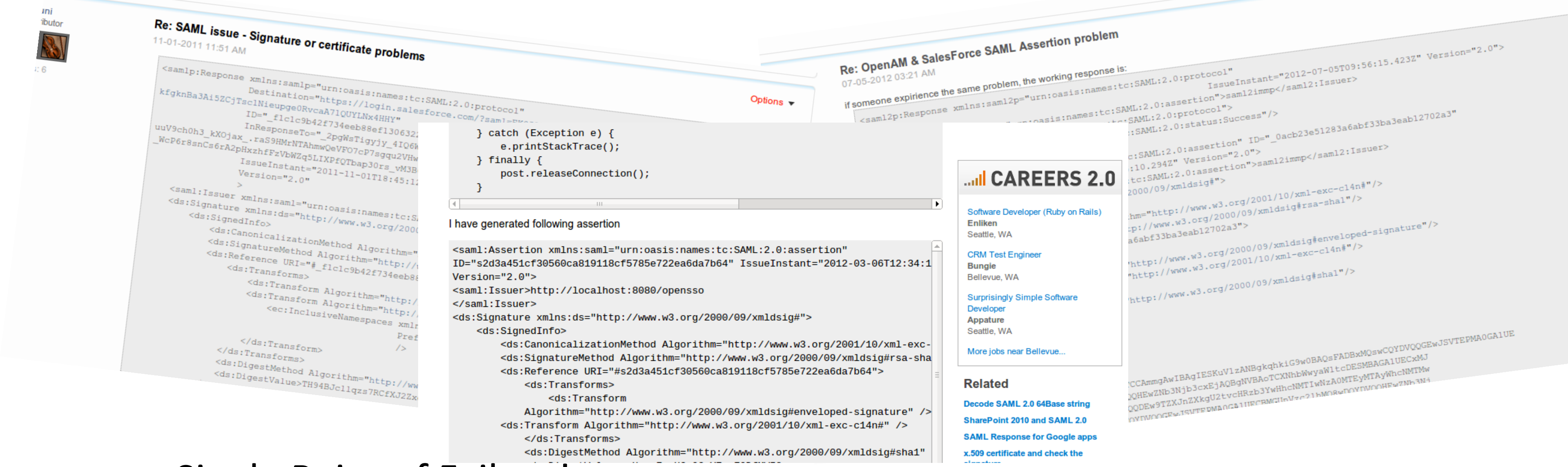
XML Signature Wrapping Attack on SAML





XML Signature Wrapping Attack on SAML – Threat model

- Change arbitrary data in the Assertion: Subject, Timestamp ...
- Attacker: everybody who can gain a signed Assertion...
  1. Registering by the Identity Provider
  2. Message eavesdropping
  3. Google Hacking



Re: SAML issue - Signature or certificate problems

11-01-2011 11:51 AM

```
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                Destination="https://login.salesforce.com/?saml=
kfgknBa3Ai5ZCjTsclNieupge0RvcaA71QUYLNx4HHY"
                ID="_f1c1c9b42f734eeb88ef130632
                InResponseTo="_2pgWsTigyjy_4IQ6W
uuV9ch0h3_kXOjax_.raS9HMrNTAhmwQeVFO7cP7sgqu2VHw
_WcP6r8snCs6rA2pHxzhfFzVbWZq5LIXPfQTbap30rs_vM3B
                IssueInstant="2011-11-01T18:45:12
                Version="2.0"
                >
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SA
  <ds:Signature xmlns:ds="http://www.w3.org/2000
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="
      <ds:SignatureMethod Algorithm="http://
      <ds:Reference URI="#_f1c1c9b42f734eeb88
        <ds:Transforms>
          <ds:Transform Algorithm="http://
          <ds:Transform Algorithm="http://
            <ec:InclusiveNamespaces xmlr
                                    Pref
                                    />
          </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://ww
        <ds:DigestValue>TH94BJcllqzs7RCfXJ2Zx
```

Options

```
    } catch (Exception e) {
       e.printStackTrace();
    } finally {
       post.releaseConnection();
    }
```

I have generated following assertion

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="s2d3a451cf30560ca819118cf5785e722ea6da7b64" IssueInstant="2012-03-06T12:34:1
Version="2.0">
<saml:Issuer>http://localhost:8080/opensso
</saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha
        <ds:Reference URI="#s2d3a451cf30560ca819118cf5785e722ea6da7b64">
            <ds:Transforms>
                <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
```

CAREERS 2.0

Software Developer (Ruby on Rails)
Enliken
Seattle, WA

CRM Test Engineer
Bungie
Bellevue, WA

Surprisingly Simple Software Developer
Appature
Seattle, WA

More jobs near Bellevue...

Related

Decode SAML 2.0 64Base string

SharePoint 2010 and SAML 2.0

SAML Response for Google apps

x.509 certificate and check the

Re: OpenAM & SalesForce SAML Assertion problem

07-05-2012 03:21 AM

if someone expirience the same problem, the working response is:

```
<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
                 IssueInstant="2012-07-05T09:56:15.423Z" Version="2.0">
      :tc:SAML:2.0:assertion">saml2immp</saml2:Issuer>
      :SAML:2.0:protocol">
      :SAML:2.0:status:Success"/>

c:SAML:2.0:assertion" ID="_0acb23e51283a6abf33ba3eab12702a3"
:10.294Z" Version="2.0">
tc:SAML:2.0:assertion">saml2immp</saml2:Issuer>
2000/09/xmldsig#">
      "http://www.w3.org/2001/10/xml-exc-c14n#"/>
hm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
p://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
a6abf33ba3eab12702a3">

"http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
"http://www.w3.org/2001/10/xml-exc-c14n#"/>
"http://www.w3.org/2000/09/xmldsig#sha1"/>

"http://www.w3.org/2000/09/xmldsig#sha1"/>

CCAmmgAwIBAgIESKuV1zANBgkqhkiG9w0BAQsFADBxMQswCQYDVQQGEwJSVTEPMA0GA1UE
QQHEwZNb3Njb3cxEjAQBgNVBAoTCXNhbWwyaW1tcDESMBAGA1UECxMJ
QQDEw9TZXJnZXkgU2tvcHRzb3YwHhcNMTIwNzA0MTEyMTAyWhcNMTMw
```

- Single Point of Failure!

XML Signature Wrapping Attack on SAML

- How about them?

Framework / Provider	Binding	Application
Apache Axis 2	SOAP	WSO2 Web Services
Guanxi	HTTP	Sakai Project (www.sakaiproject.org)
Higgins 1.x	HTTP	Identity project
IBM Datapower XS40	SOAP	Enterprise XML Security Gateway
JOSSO	HTTP	Motorola, NEC, Redhat
WIF	HTTP	Microsoft Sharepoint 2010
OIOSAML	HTTP	Danish eGovernment (e.g. www.virk.dk)
OpenAM	HTTP	Enterprise-Class Open Source SSO
OneLogin	HTTP	Joomla, Wordpress, SugarCRM, Drupal
OpenAthens	HTTP	UK Federation (www.eduserv.org.uk)
OpenSAML	HTTP	Shibboleth, SuisseID
Salesforce	HTTP	Cloud Computing and CRM
SimpleSAMLphp	HTTP	Danish e-ID Federation (www.wayf.dk)
WSO2	HTTP	eBay, Deutsche Bank, HP

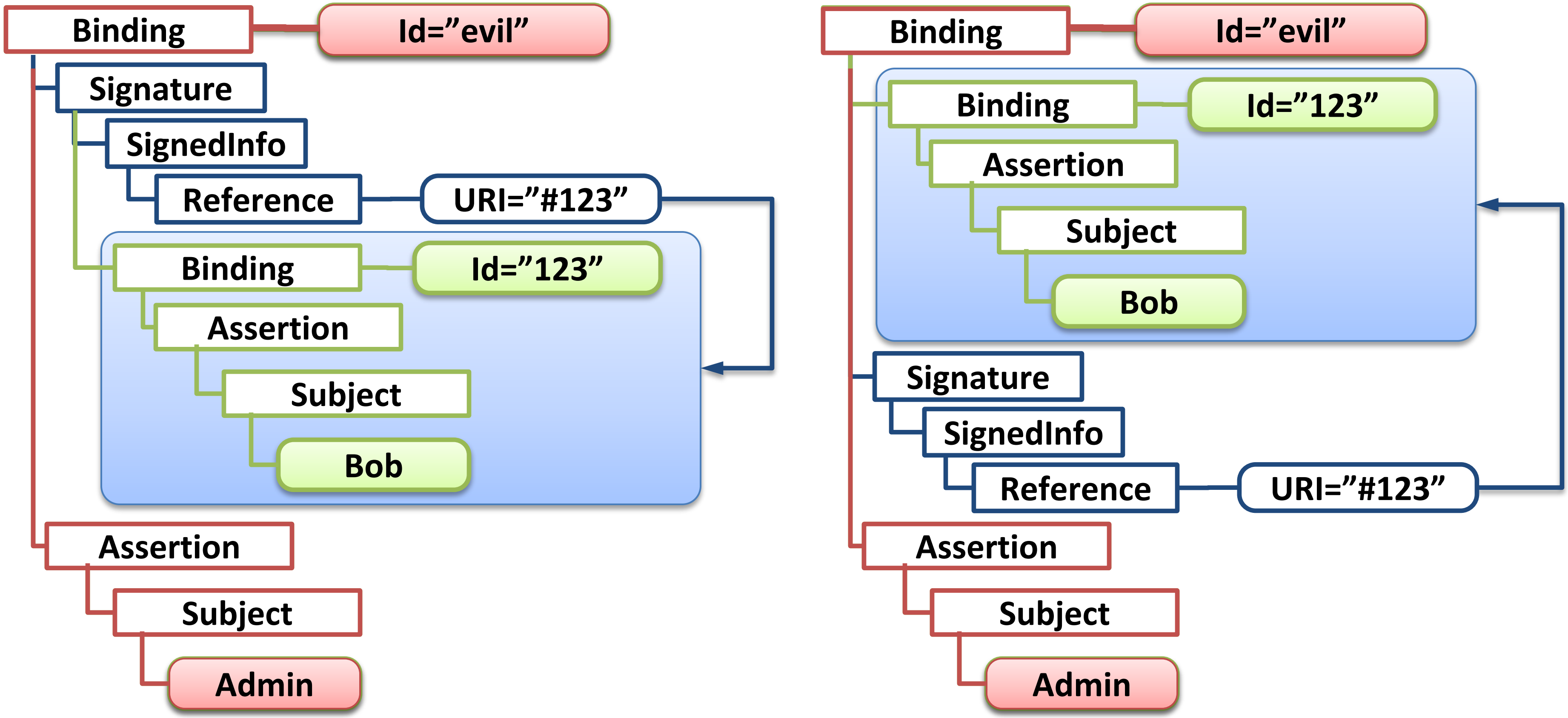
Overview

1. Securing SAML with XML Signature
2. XML Signature Wrapping Attacks
3. Practical Evaluation
4. Penetration Test Library
5. Countermeasures
6. Conclusion

XML Signature Wrapping Attack on SAML – Results

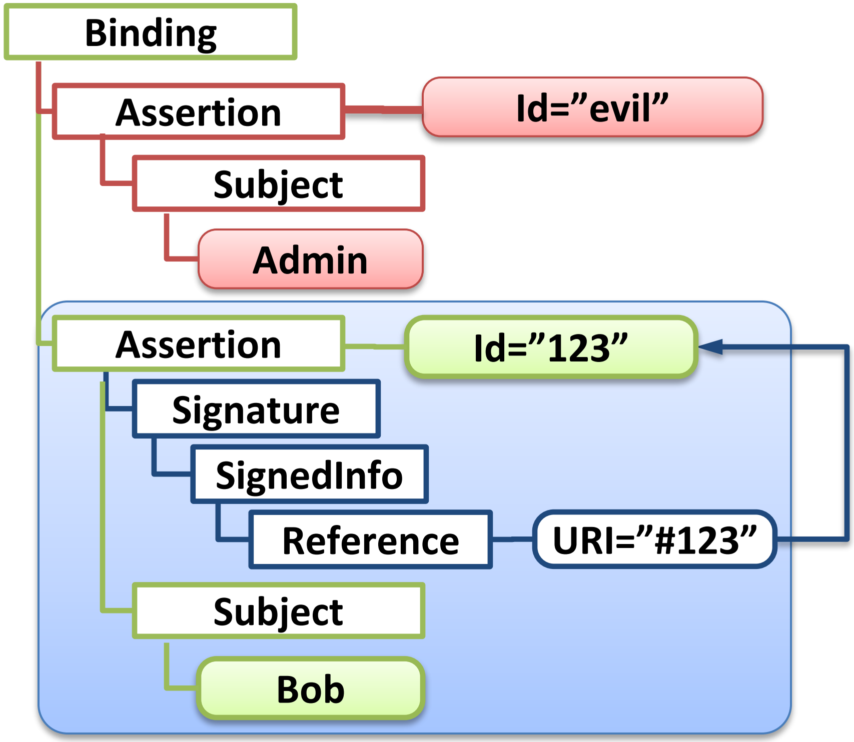
Guanxi, JOSSO

WSO2

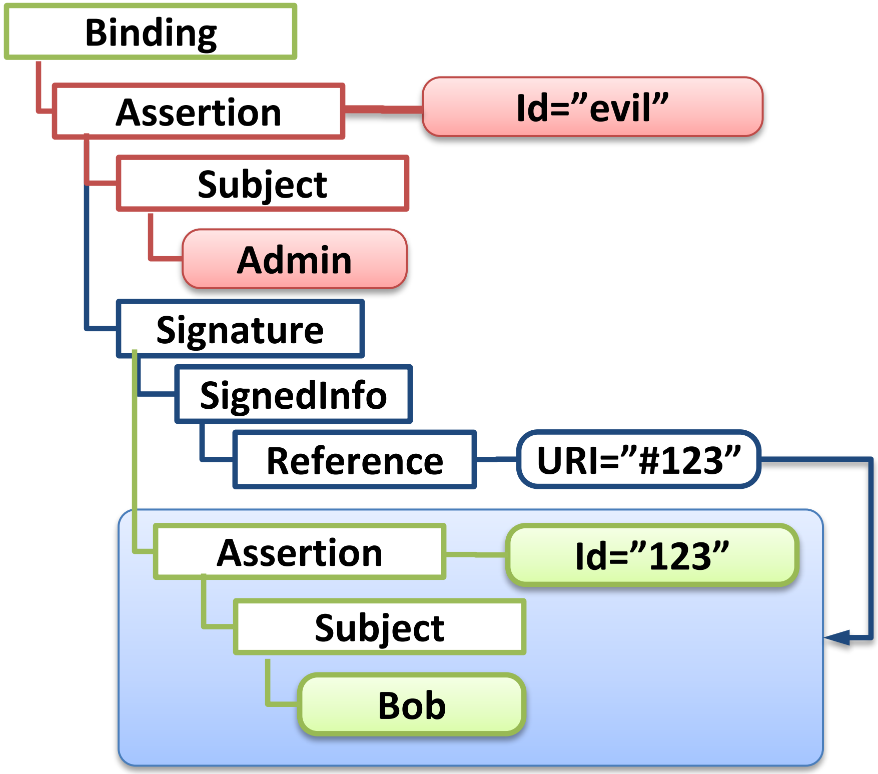


XML Signature Wrapping Attack on SAML – Results

Higgins, Apache Axis2, IBM XS 40



OpenAM, Salesforce



Attack on OpenSAML

- Is Signature Wrapping always that easy?

- OpenSAML implemented a few countermeasures:
  1. Checked if the signed assertion has the same ID value as the processed one
  2. Validated XML Schema
     - Not possible to insert two elements with the same ID values

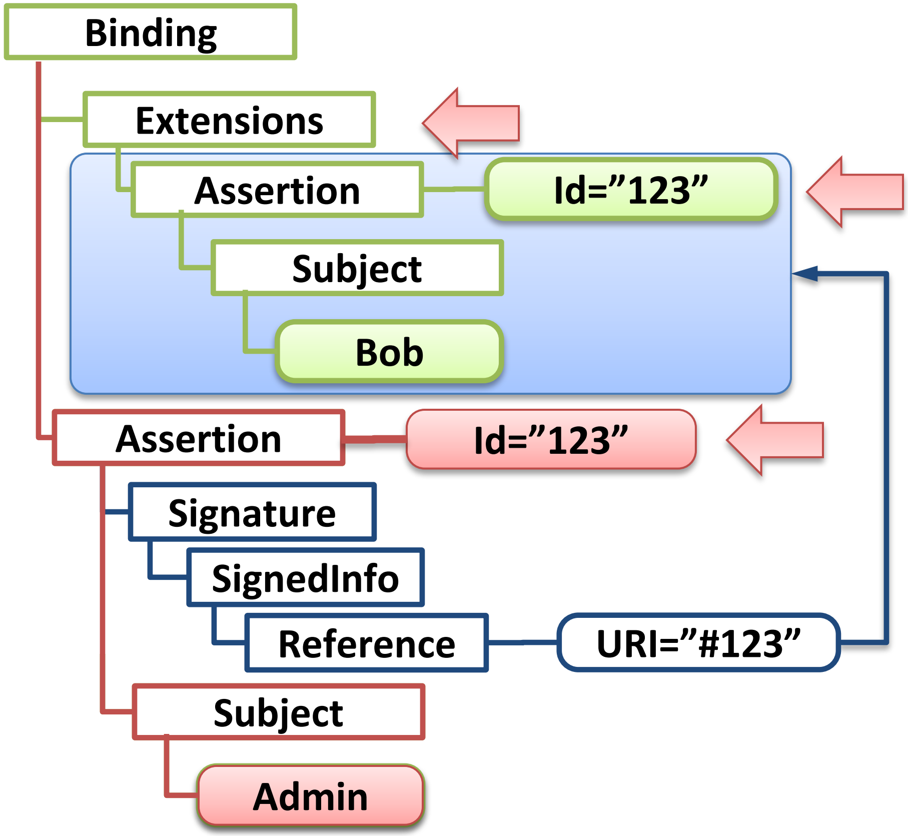
Attack on OpenSAML

1. ID values checking: Basic idea – using two identical ID values ✓
2. XML Schema validation:
   1. Put the Assertion into an extensible element (e.g. <Extensions>) ✓
   2. Two identical ID attributes (XML Xerces Parser bug) ✓

- Which element is verified?

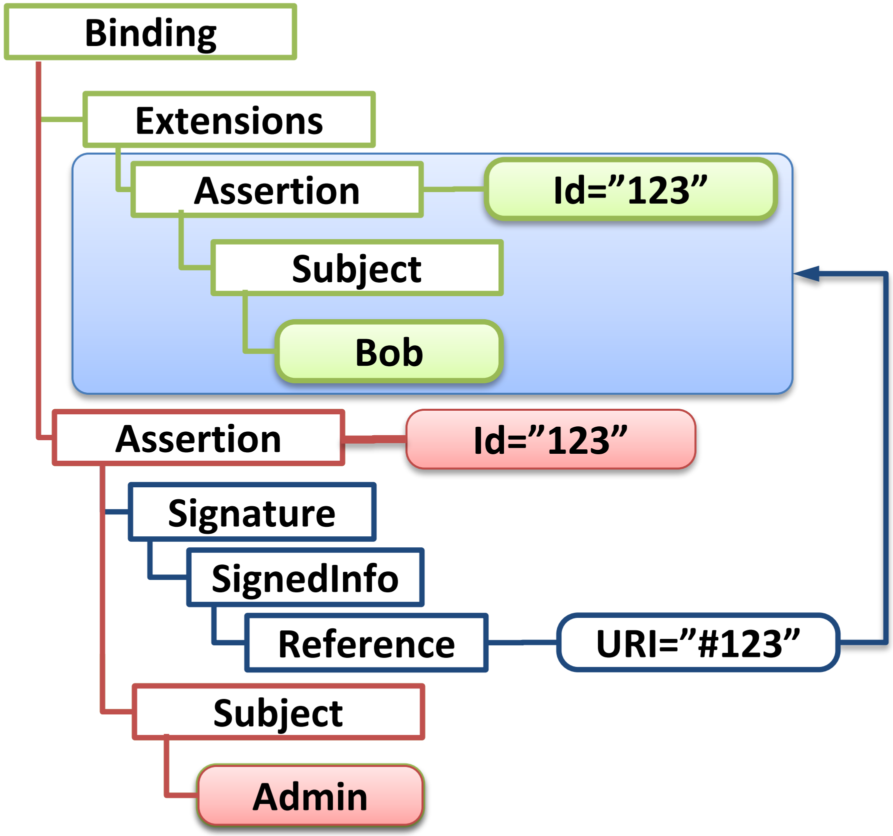
C++ takes the first found element

OpenSAML C++

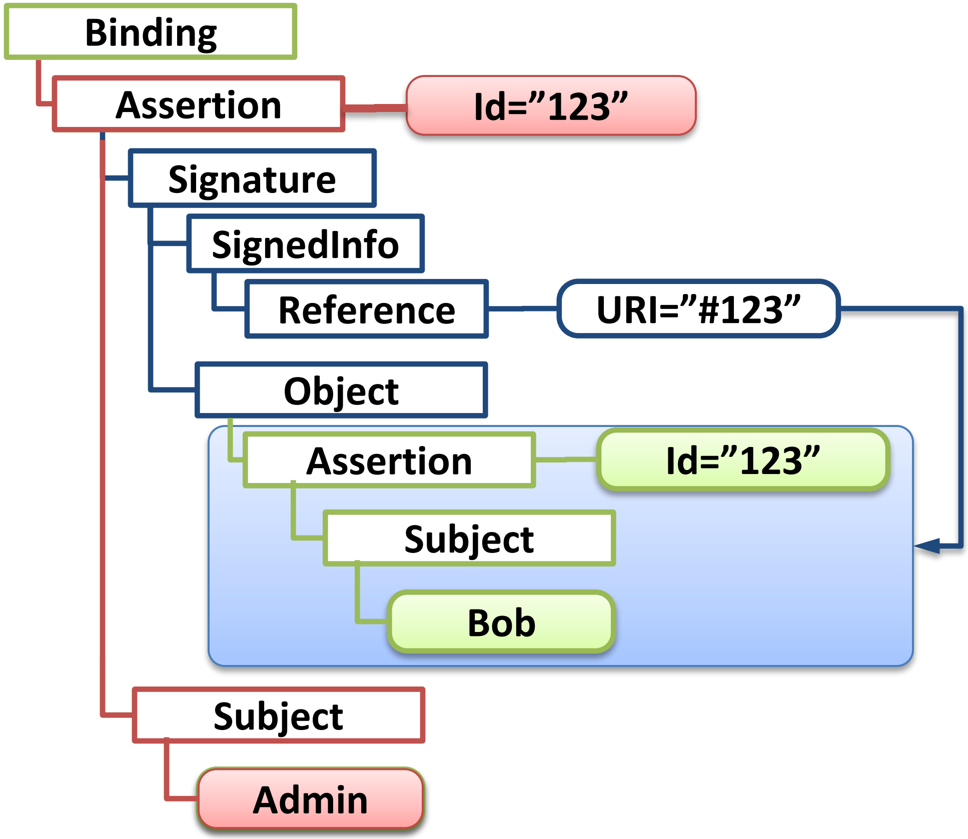


Attack on OpenSAML

OpenSAML C++ references the **first** found element

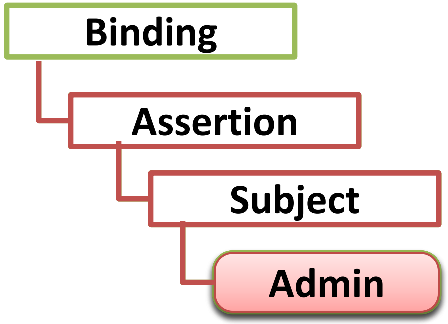


OpenSAML Java references the **last** found element



Beyond Signature Wrapping: Signature Exclusion

- Lame but …



- …Worked against:
  - Apache Axis2
  - JOSSO
  - OpenAthens

SAML Signature Wrapping – Summary

Framework / Provider	Signature Exclusion	Signature Wrapping
Apache Axis 2	X	X
Guanxi		X
Higgins 1.x		X
IBM Datapower XS40		X
JOSSO	X	X
WIF		
OIOSAML		X
OpenAM		X
OneLogin		X
OpenAthens	X	
OpenSAML		X
Salesforce		X
SimpleSAMLphp		
WSO2		X

Enterprise Applications

Danish eGovernment

Joomla, Wordpress, SugarCRM, Drupal

Shibboleth, SwissID …

✓

Overview

1. Securing SAML with XML Signature
2. XML Signature Wrapping Attacks
3. Practical Evaluation
4. Penetration Test Library
5. Countermeasures
6. Conclusion

Penetration Test Library

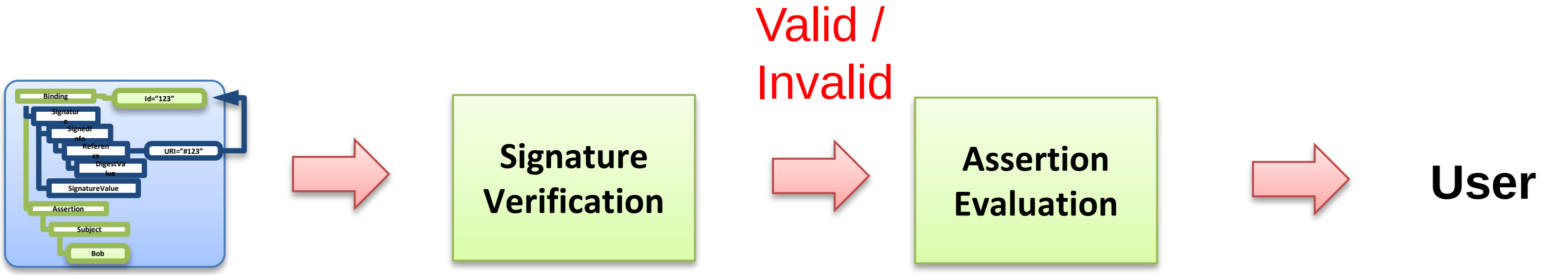
- Considered all the attack vectors:
  1. Different permutations of signed / processed Assertions
  2. Id processing
  3. Signature exclusion attacks
  4. XML Schema extensions
- Further attacks on Salesforce interface
- Will be included in our WS-Attacker framework
  - http://ws-attacker.sourceforge.net/

Overview

1. Securing SAML with XML Signature
2. XML Signature Wrapping Attacks
3. Practical Evaluation
4. Penetration Test Library
5. Countermeasures
6. Conclusion

Countermeasures

- General problem: different processing modules have different views on documents



Signature Verification

Valid / Invalid

Assertion Evaluation

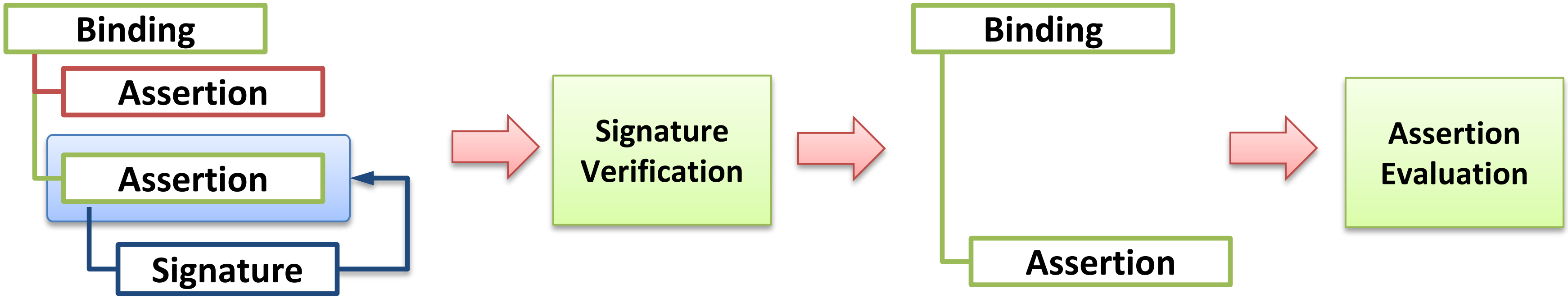
User

Id-based

/Binding/Assertion/Subject

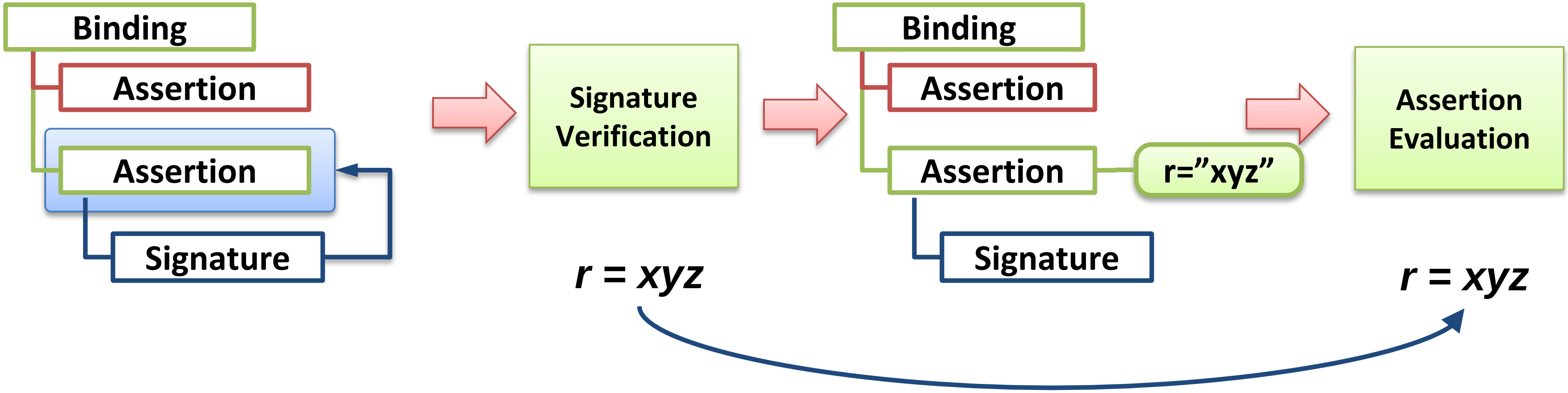
Countermeasure 1: Strict Filtering

- Forward only signed elements
- Also called *see-only-what-is-signed*



Countermeasure 2: Data Tainting

- Signature verification generates a random number ***r***
- The verified data is tainted with ***r***
- ***r*** is forwarded to the Assertion evaluation logic



Overview

1. SAML Assertion
2. Securing SAML with XML Signature
3. XML Signature Wrapping Attacks
4. Practical Evaluation
5. Countermeasures
6. Conclusion

Conclusion

- We showed critical Signature Wrappings in SAML, 12 out of 14 frameworks affected!
- All providers informed
- Signature Wrapping known since 2005, but:
  - Not in focus of research community
  - Nearly all implementations are vulnerable
  - Not easy to fix: many permutations, vulnerable libraries
- Be aware of Signature Wrapping when applying:
  - In Web Services
  - SAML
- Beyond XML: Could be applied in all the scenarios where different processing modules have different views on documents

Thank you for your attention

Juraj Somorovsky[1], Andreas Mayer[2], Jörg Schwenk[1], Marco Kampmann[1], and Meiko Jensen[1]

[1]Horst-Görtz Institute for IT-Security, Ruhr-University Bochum

[2]Adolf Würth GmbH & Co. KG