On breaking SAML

Be Whoever You Want to Be

von David Foerster

Gliederung

- Übersicht & Motivation
- XML Signature Wrapping Attacks
  - Vorstellung
  - Gegenmaßnahmen
- Zusammenfassung

Übersicht & Motivation

- SAML
  - Übersicht
  - Motivation
  - Geschichte
- XML Signature
  - Beispiele
- XML Schema Definition

SAML – Übersicht

- **S**ecurity **A**ssertion **M**arkup **L**anguage
  - 2002 standardisiertes (OASIS),
  - zustandsloses,
  - XML-basiertes Austauschformat
- besteht aus Aussagen (*assertions*), die Fakten bzgl. eines Subjekt beschreiben:

  Aussageform A wurde geprüft zur Zeit t von Prüfer R bezüglich Subjekt S unter der Bedingung C.[1]

  - Authentifizierungsaussage
  - Attributsaussage
  - Autorisierungsentscheidungsaussage

SAML – Motivation

- Authentifizierungs- und Authorisierungsinformationen zwischen
  - Identitätsanbietern (z. B. single sign-on) und Dienstanbietern,
  - an verteilten Prozessen beteiligten Systemen
- üblicherweise
  - als Web Services angeboten
  - von anderen Web Services genutzt

SAML – Geschichte

- kaum Unterschiede zwischen v1.0 und v1.1
- SAML 2.0 (2005)
  - inhaltliche Obermenge von v1.1
  - sprachlich inkompatibel zu v1.1
  - beschreibt zusätzlich
    - zur Identitätsprüfung eingesetztes Verfahren,
    - eingesetztes Authentifizierungssystem
    - sonstige zugehörige Richtlinien
  - zusammengeführt mit Liberty Foundation Federation Framework (ID-FF)

XML Signature (1)

- Signierung von kanonisierbaren Ressourcen
  - nicht als Bytestrom
  - sondern als logische Entität
    - Beispiel: XML-Dokumente haben mehrere gleichwertigen Binärdarstellungen
- unter Angabe von
  - MIME type
  - Kanonisierungsmethode
  - Digest und Digest-Methode
  - Signaturaussteller (wenn nicht aus Kontext abgeleitet)

XML Signature (2)

- Angabe der signierten Element durch Referenz
- mögliche Lagen zur signierten Element E:
  - Element ist in Signatur eingebettet (*enveloped*)
  - Signatur ist in Element eingebettet (*enveloping*)
  - ohne Eltern-Kind-Beziehung (*detached*)
- große Anzahl an Kombinationen aus
  - Kanonisierungsmethoden,
  - Digest-Methoden und
  - Lagenbeziehungen [sic!]
- macht korrekte Implementierung im Kontext schwierig

XML Signature: Beispiel 1

```
<Signature ID="signature-example"
  xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <SignatureMethod Algorithm="…/xmldsig#rsa-sha1" />
    <Reference URI="#object">
      <DigestMethod Algorithm="…/xmldsig#sha1" />
      <DigestValue>OPnpF/ZNLDxJ/I …</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>nihUFQg4mDhLgec …</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <RSAKeyValue>
        <Modulus>4IlzOY3Y9fXoh3Y5f06wB …</Modulus>
        <Exponent>AQAB</Exponent>
      </RSAKeyValue>
    </KeyValue>
  </KeyInfo>
  <Object ID="object">some tex
    with spaces and CR-LF.</Object
</Signature>
```

XML Signature: Beispiel 2

```
<soap:Envelope>
  <soap:Header> … </soap:Header>
  <soap:Body>
    <saml:Assertion> … </saml:Assertion>
  </soap:Body>
  <Signature ID="signature-example"
    xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
     <CanonicalizationMethod
       Algorithm="…/REC-xml-c14n-20010315" />
      …
      <Reference URI="">
        <DigestValue>OPnpF/ZNLDxJ/I …</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>nihUFQg4mDhLgec …</SignatureValue>
    <KeyInfo> … </KeyInfo>
  </Signature>
</soap:Envelope>
```

XML Schema Definition

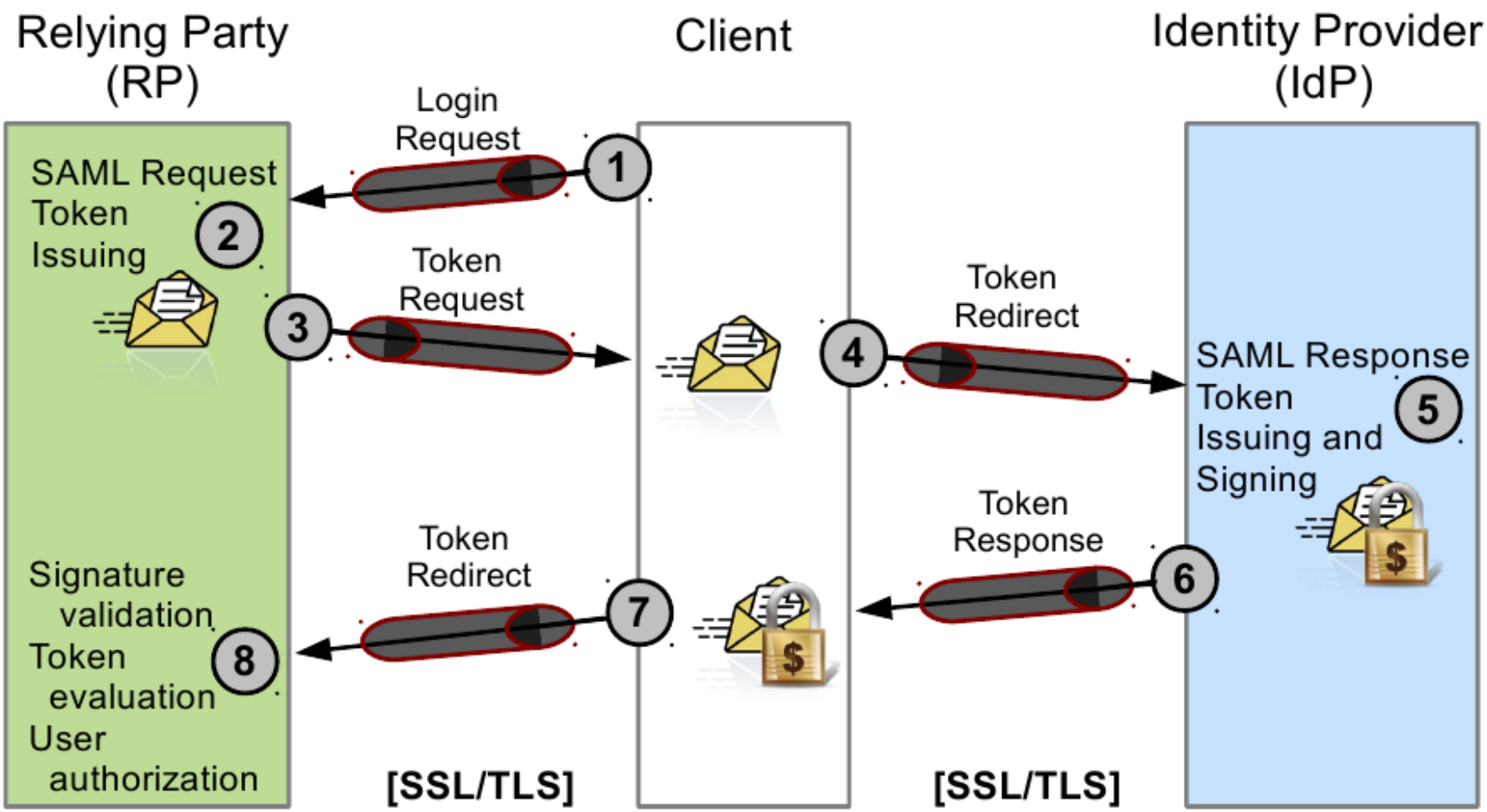
- definiert Struktur eines XML-Dokuments
- erlaubt automatische Konformitätsprüfung

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="Address">
    <complexType>
      <sequence>
        <element name="Recipient" type="string" />
        <element name="Street" type="string" />
        <element name="Town" type="string" />
        <element name="Country" minOccurs="0">
          <simpleType>
            <restriction base="string">
              <enumeration value="DE" />
              <enumeration value="US" />
            </restriction>
          </simpleType>
        </element>
      </sequence>
    </complexType>
  </element>
</schema>
```

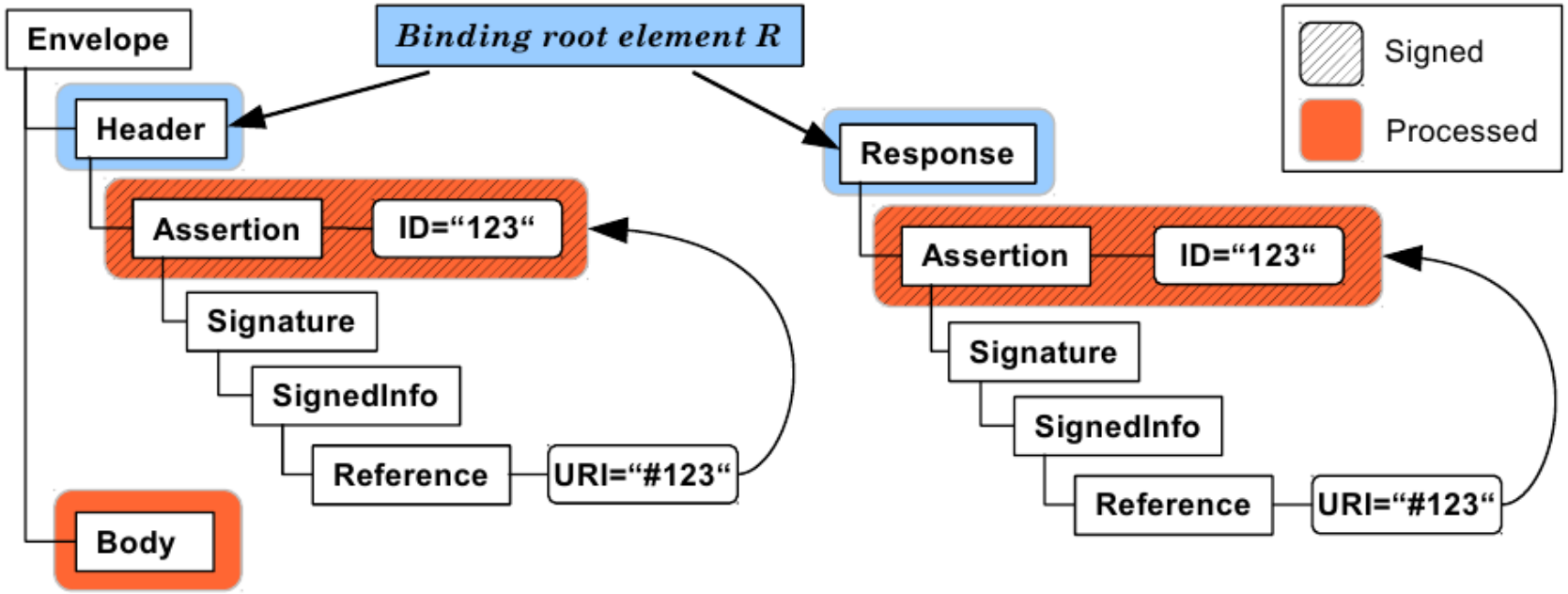
Gliederung

- Single-Sign-On-Szenarium
- Aufgaben der Relying Party
- Angriffsvoraussetzung
- Angriffspunkt
- XML Signature Wrapping Attacks
- Einfache Angriffe
- Verfeinerte Angriffe
- Angriffe – Rückschau
- Gegenmaßnahmen

Single-Sign-On-Szenarium



Aufgaben der Relying Party

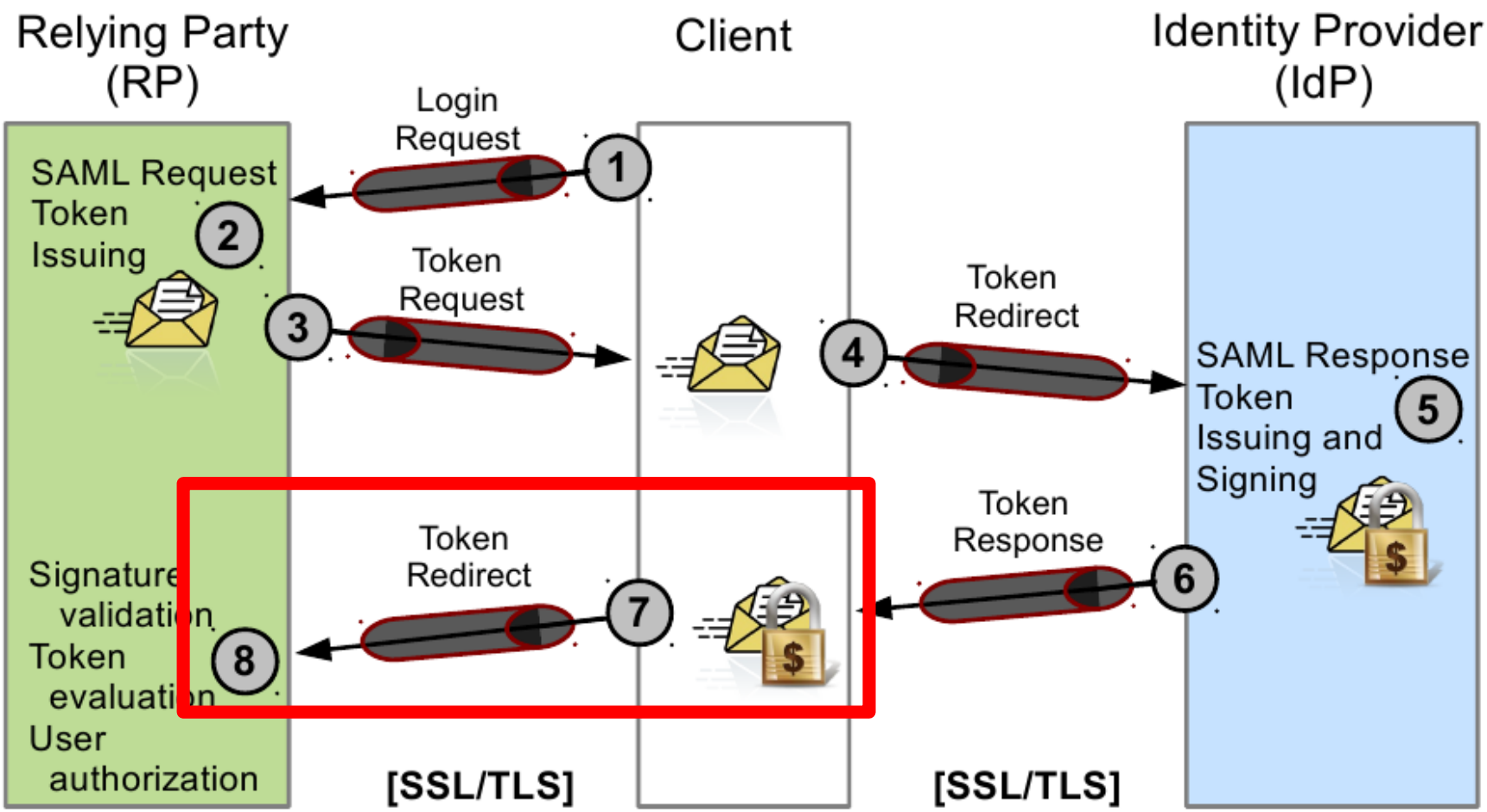


1. XML-Schema validieren
2. Vertrauenswürdigkeit des Signaturausstellers
3. Signatur des SAML-Objekts prüfen
4. SAML-Objekt verarbeiten

Angriffsvoraussetzung

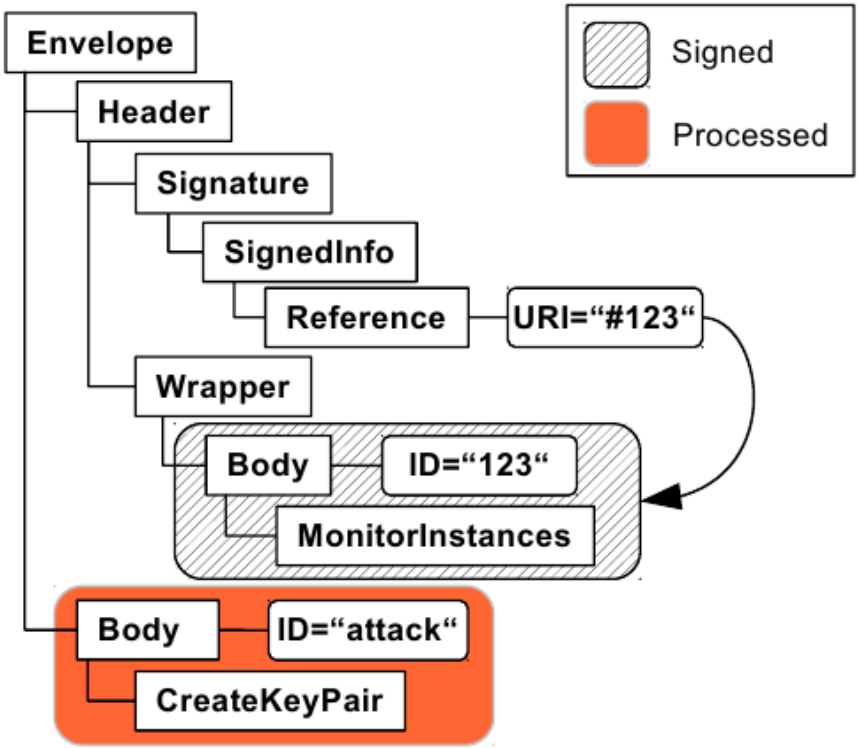
- ein „gültiges“ SAML-Token
  - von ihm selbst generiert bei Identity Provider,
  - Verbindung belauschen,
  - aus Browser- oder Proxy-Cache oder
  - aus technischen Foren
- ein verwundbares Ziel (Relying Party)
- mehr nicht

Angriffspunkt



XML Signature Wrapping Attacks

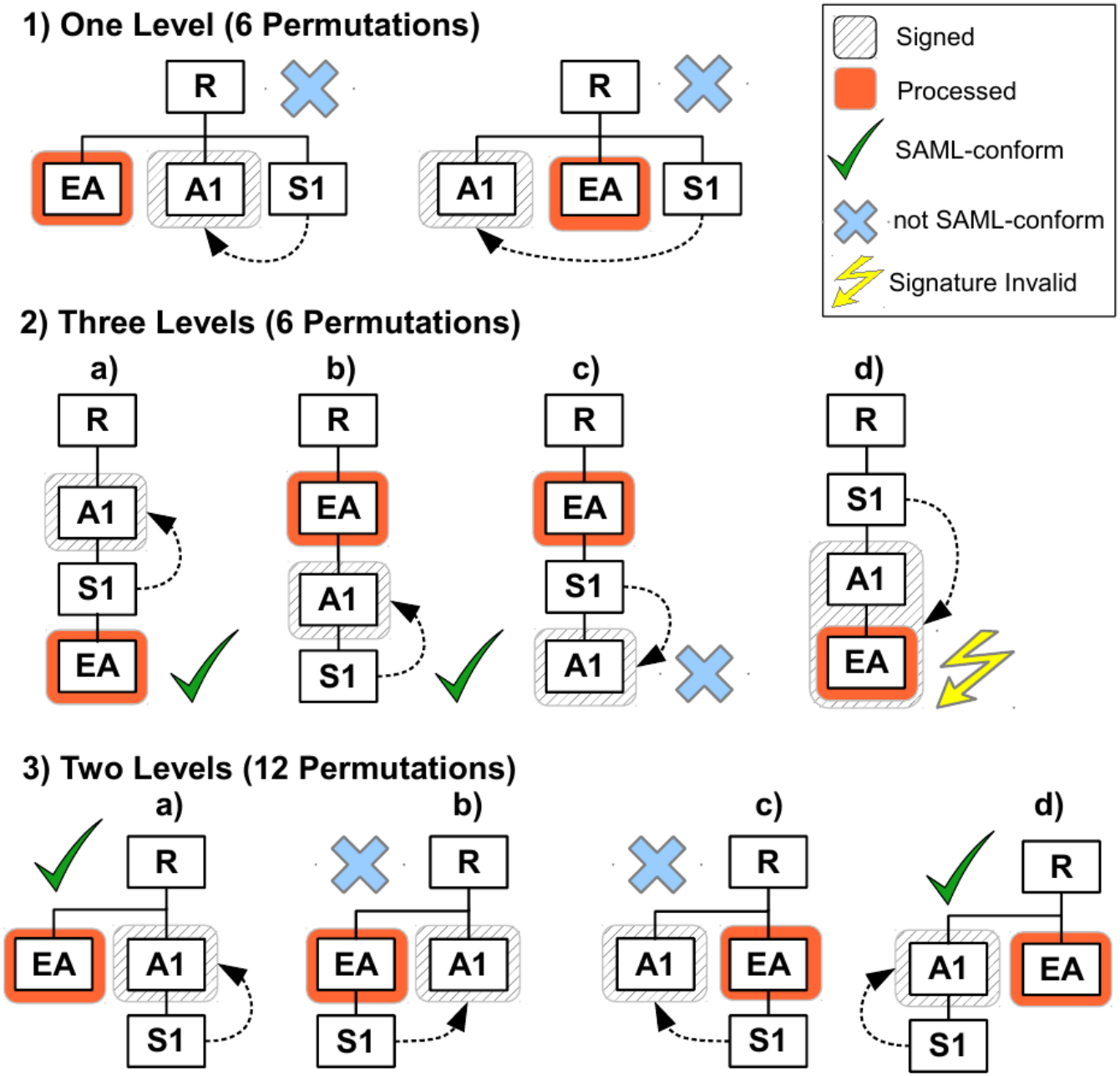
- verschiedene Sichten auf Dokument von
  - formaler Prüfung (1–3) und
  - Geschäftslogik (4)
- Angreifer injiziert gefälschte Elemente
  - von Prüfung ignoriert
  - aber von Geschäftslogik verwendet



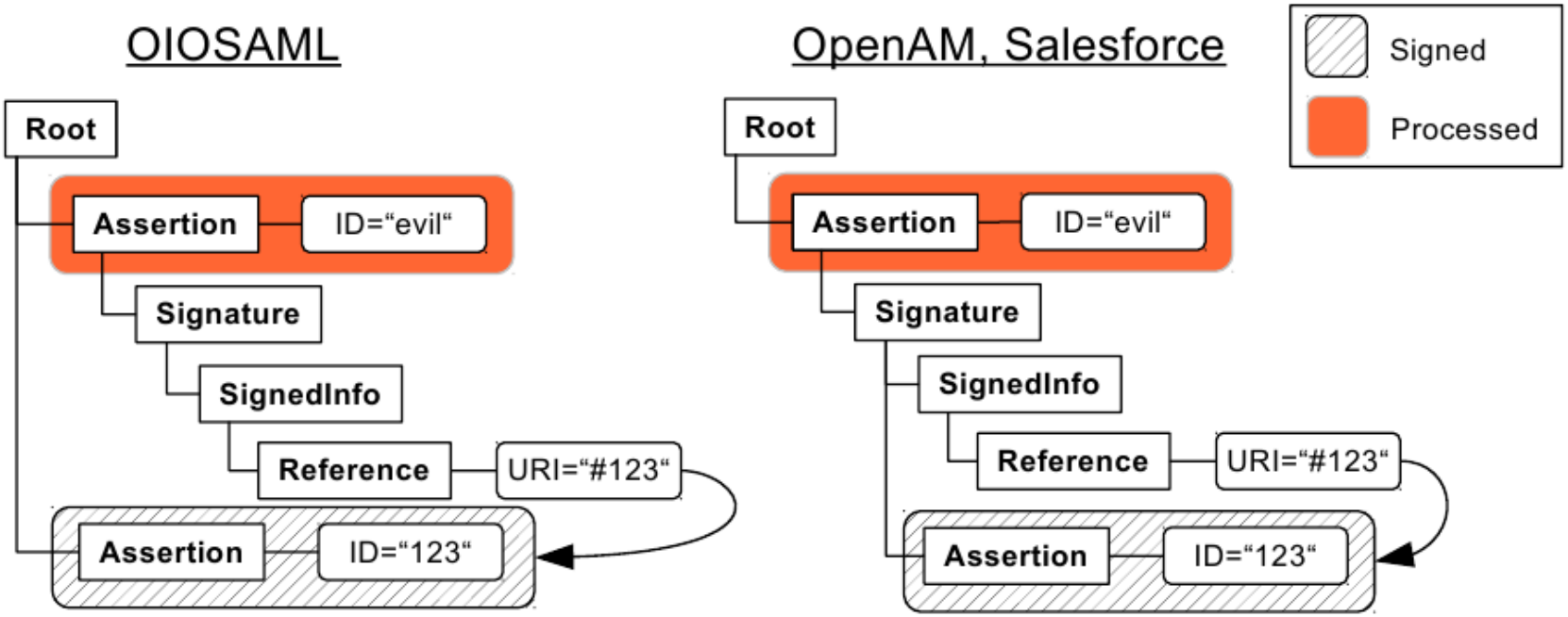
Einfache Angriffe

- signature exclusion attacks
  - gefälschtes Element unsigniert übermitteln
  - falls keine Signatur erzwungen
  - 3/14 Implementierungen verwundbar

Verfeinerte Angriffe (1)

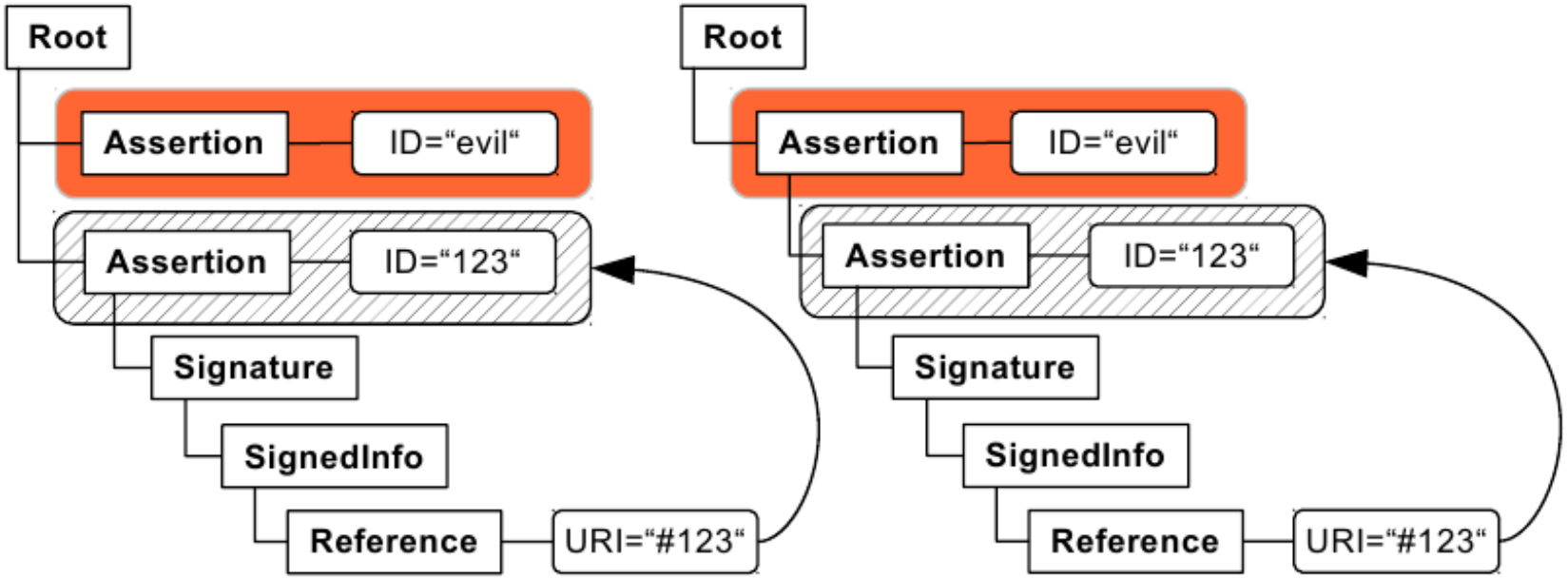


Verfeinerte Angriffe (2)

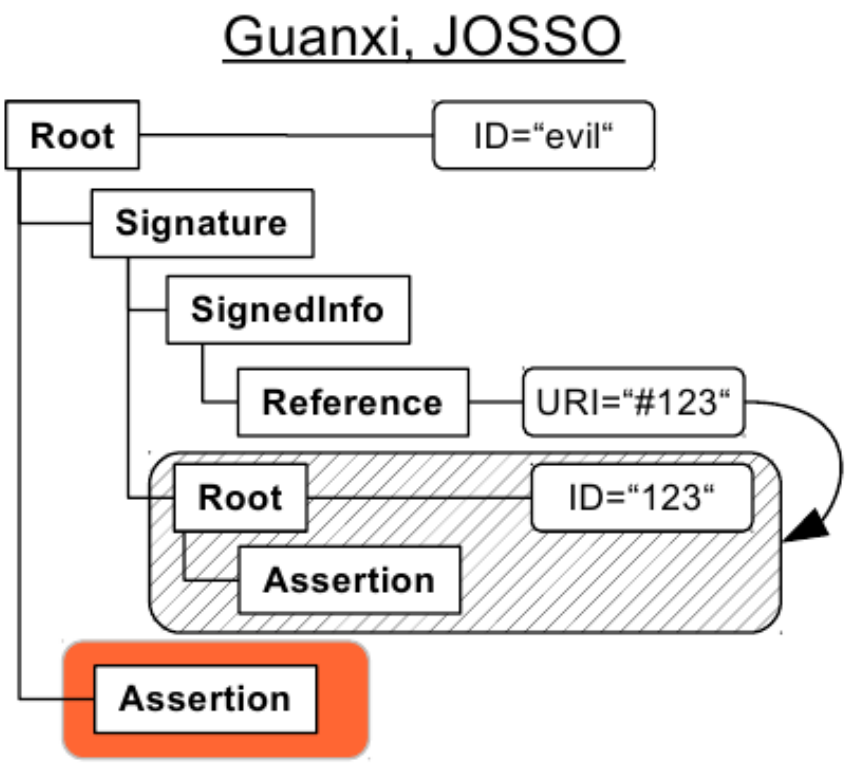


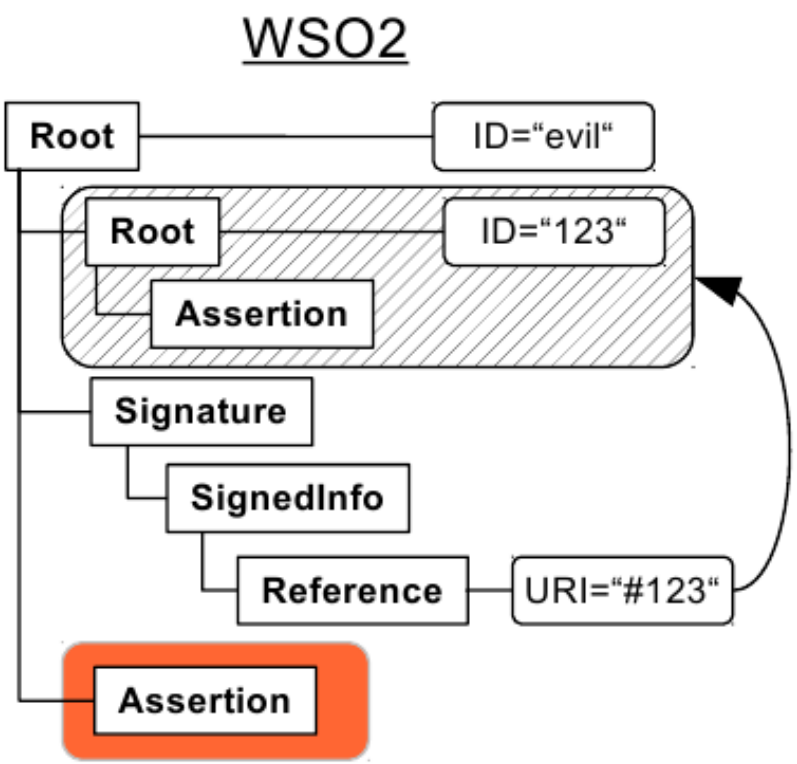
Verfeinerte Angriffe (3)

Higgins, Apache Axis2, IBM XS 40 Security Gateway



Verfeinerte Angriffe (4)





Angriffe – Rückschau

1. XML-Schema validieren
2. Vertrauenswürdigkeit des Signaturausstellers
3. Signatur des SAML-Objekts prüfen
4. SAML-Objekt verarbeiten

- Auslassung oder schlechte Integration mindestens eines Schritts
  - 12/14 Implementierungen verwundbar
  - Blüten:
    - hart kodierte XML-Pfade
    - mehrdeutige ID zulässig

Gegenmaßnahmen

- Tilge Dokumentteile falls
  - missgebildet,
  - unsigniert,
  - falsch signiert oder
  - ohne Vertrauen gegenüber Signierendem
  - *und* ohne gültig signierte Kinder
- Reichere Dokument an
  - zusätzliches, nur intern verwendetes Attribut für gültig signierte Teilbäume
  - an SAML-Verarbeitung weiterreichen
  - verarbeitet nur Teile mit solchem Attribut

Zusammenfassung

- viele Teilkomponenten, hohe Integrationskomplexität
- inkonsistente Dokumentensicht seitens Modulen
- große Mehrheit der SAML-Implementierungen verwundbar
  - selbst 6 bzw. 9 Jahre nach Standardisierung
  - trotz gereifter XML-Frameworks
- niedrige Voraussetzungen an erfolgreiche Angriffe

Fragen / Diskussion

?

Gliederung

- Übersicht & Motivation
  - SAML
  - XML Signature
  - XML Schema Definition
- XML Signature Wrapping Attacks
  - Vorstellung
  - Gegenmaßnahmen
- Zusammenfassung

Quellen

- Somorovsky, Mayer, Schwenk, Kampmann und Jensen, „On Breaking SAML: Be Whoever You Want to Be“. *21st USENIX Security Symposium Proceedings* [pdf]
- Wikipedia
  - XML Schema Definition
  - XML Signature
  - Security Assertion Markup Language