IMPLEMENTASI XML FIREWALL UNTUK KEAMANAN WEB SERVICE

Tawar, Jazi Eko Istiyanto
Universitas Ahmad Dahlan, Universitas Gadjah Mada
Yogyakarta
tawar@telkom.net, jazi@ugm.ac.id

Abstrak

Web services sedang dipromosikan sebagai teknologi terkini untuk membangun sistem terdistribusi berbasis web. Teknologi web services memungkinkan sistem yang dikembangkan akan mempunyai kekuatan khusus, yaitu aplikasi jaringan yang dikembangkan akan bersifat interoperabel dan mudah dikembangkan dengan memanfaatkan komponen yang sudah ada. Web services memiliki karakteristik ***netral bahasa*** *dan* ***netral platform,*** *sehingga mengakibatkan munculnya tantangan dan ancaman keamanan yang baru . Mekanisma yang mungkin untuk melakukan pengamanan web service adalah menggunakan XML Firewall, yaitu modul keamanan yang merupakan gabungan protokol-protokol keamanan yang direkomendasikan oleh W3C dan OASIS.*

Keywords

SOAP, XML, XML firewall, Keamanan web services.

1. Pendahuluan

Web services sedang dipromosikan sebagai teknologi terkini untuk membangun sistem terdistribusi berbasis web [1], [2]. Sistem yang dibangun dengan memanfaatkan web service akan mempunyai kekuatan khusus, yaitu aplikasi jaringan yang dikembangkan bersifat interoperabel dan mudah dikembangkan dengan memanfaatkan komponen yang ada [2].

Banyak aspek web service yang hingga saat ini masih dalam pengembangan, diantaranya aspek bisnis, aspek keamanan dll. Jika dilihat dari aspek keamanan, sejumlah layanan jaringan diketahui memiliki resiko keamanan. Sebagai salah satu layanan jaringan, web service juga memiliki resiko keamanan. Salah satu resiko keamanan pada web service adalah terbukanya port yang digunakan untuk komunikasi.

Pengamanan jaringan sering menggunakan firewall. Mekanisma yang mungkin untuk melakukan pengamanan web service adalah menggunakan firewall yang bekerja pada *content level* yaitu dengan mengimplementasikan **XML Firewall**. XML firewall bekerja dengan cara melakukan pemeriksaan terhadap message yang diterima, sehingga hanya *message* yang sah saja yang dapat mengakses web service [3], [4].

2. Web Service

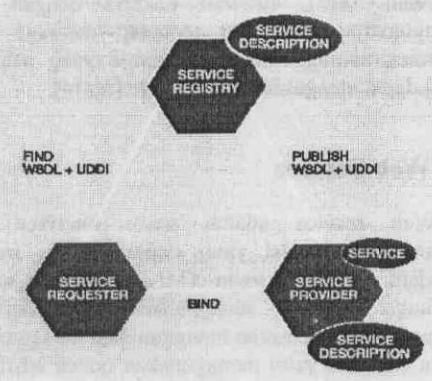
Web service adalah suatu interface bagi sekumpulan operasi yang dapat diakses melalui standard messaging yaitu XML. Suatu web service berfungsi untuk menjalankan tugas tertentu. *Property*-nya dijelaskan menggunakan menggunakan suatu standard yaitu menggunakan notasi XML dan selanjutnya disebut dengan *service description* [5].

Teknologi *web service* merupakan bentuk baru dari teknologi yang sudah cukup tua yaitu RPC *Remote Procedure Call* [6]. RPC (Remote Procedure Call) sering dipakai untuk komunikasi antar aplikasi terdistribusi. Protokol lain yang juga sering dipakai antara lain DCOM, RMI dan CORBA tidak bisa diterima sebagai protokol umum serta tidak menyediakan standar mekanisma untuk dipakai melalui internet. Beberapa perusahaan besar seperti IBM, Microsoft, Oracle , Sun dan SAP telah menerima web service sebagai sebuah protokol standard. Hal ini mengakibatkan web service menjadi sebuah *framework* untuk komputasi terdistribusi dan mempunyai potensi untuk sukses [7].

3. Arsitektur Web Service

Web service merupakan aplikasi modular yang dapat dijalankan melalui internet. Arsitektur web service terdiri dari tiga buah role dan tiga operasi dasar. Ketiga role itu adalah **service provider**, **service requester** dan **service registry**. Sedangkan ketiga operasi dasar itu adalah yaitu : **publish, find** dan **bind**. Obyek yang ada adalah **service** dan **service description**. Secara visual, arsitekturnya dapat dijelaskan dalam gambar 1.

Pada gambar 1, service provider membuat web service beserta definisinya kemudian bersama service registry mempublikasikannya menggunakan suatu standar yang disebut spesifikasi **(UDDI) Universal Description, Discovery and Integration.** Setelah service dipublikasikan, service requester akan menemukan service tersebut *melalui interface* UDDI. Registry UDDI memfasilitasi service requester dengan service description berbasis WSDL (Web Service Description Language) dan URL yang *menunjuk lokasi service tersebut. Service* requester akan memanfaatkan informasi ini untuk secara langsung mengikat (bind) dan memanfaatkan service tersebut. Semua komunikasi yang terjadi ini *dilakukan* melalui suatu protokol standar untuk



Gambar 1. Arsitektur web service

messaging yaitu SOAP (Simple Object Access Protocol) [5].

3.1 SOAP

Simple Object Access Protocol (SOAP) adalah protokol berbasis XML yang digunakan untuk pertukaran informasi antar aplikasi pada sistem terdistribusi [8]. SOAP dapat dipakai digunakan dalam berbagai sistem messaging dan dapat melewati berbagai protokol transport, tetapi perhatian utama SOAP adalah RPC (Remote Procedure Call) melalui HTTP.

Message SOAP berisi definis struktur message dan aturan-aturannya, digunakan untuk sistem *messaging* satu arah untuk melakukan Remote Procedure Call (RPC) dengan pola dialog request-response [9].

SOAP bersifat *platform independent,* tidak terikat pada satu platform, sehingga memungkinkan komunikasi antar aplikasi yang beragam [7].

3.2 Anatomi message SOAP

Seperti dijelaskan sebelumnya bahwa message SOAP berupa dokumen *XML, dokumen ini bersifat* well-formed XML [10]. Sintaks message SOAP secara detail dijelaskan di URL http://www.w3.org/TR/SOAP/. Berikut adalah komponen *message secara garis besar :*

- Deklarasi XML (opsional)
- **Envelope** (elemen root), terdiri dari : **Header** (opsional) dan **body** (bersifat wajib)

Envelope merupakan elemen puncak dari dokumen, berisi kerangka kerja yang menyatakan **apa** yang ada dalam message, **siapa** yang boleh menggunakan.

Header, mekanisma fleksibel untuk memberikan fitur-fitur tambahan bagi message SOAP. Contoh fitur yang dapat diimplemantasikan menggunakan elemen header adalah otentikasi, manajemen transaksi, pembayaran dll. Elemen header merupakan elemen anak dengan posisi pertama.

Body, berisi informasi yang akan disampaikan pada penerima. SOAP mendefinisikan satu element untuk body, yang merupakan elemen *Fault* digunakan untuk melaporkan error.

Contoh message SOAP :

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
    http://schemas.xmlsoap.org/soap
    /envelope/
  xmlns:ds="http://www.w3.org/2000/
    09/xmldsig#">
  <SOAP-ENV:Header>
    <ds:Signature>
      <ds:SignedInfo>
      .....
      </ds:SignedInfo>
      <ds:SignatureValue>
      .....
      </ds:SignatureValue>
      <ds:KeyInfo>
      .....
      </ds:KeyInfo>
    </ds:Signature>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <s:MyServices
    xmlns:s="http://MyURL/service/">
      .....
    <!--Parameters call-->
      .....
    </s:MyServices>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.3 WSDL

Web Service Description Language (WSDL) menyajikan komponen *service description* di dalam arsitektur web service, didalamnya termasuk menjelaskan web service yang dapat digunakan oleh client. WSDL digunakan untuk menyajikan interface

publik bagi web service menggunakan format XML. Interface publik ini bisa mengandung informasi pada seluruh function, informasi tipe data, informasi binding, informasi alamat dari suatu service. WSDL juga memiliki built-in extention untuk menyajikan service berbasis SOAP [11],[7].

3.4 UDDI

Universal Description, Discovery and Integration (UDDI) menyajikan komponen *service discovery* di dalam arsitektur web service. UDDI mengandung dua komponen utama yaitu , **pertama**, UDDI adalah *spesifikasi teknis untuk membangun direktori* bisnis terdistribusi dan web service. Data disimpan dalam format XML, dan spesifiaksi UDDI mengandung API untuk pencarian data dan publikasi data baru. **Kedua**, UDDI Business Registry adalah implemetasi penuh dari spesifikasi UDDI. Dengan UDDI dimungkinkan sembarang perusahaan mendaftarkan diri beserta layanan-layanannya [7].

4. Aspek Keamanan Web Service

Pertukaran informasi antar perusahaan melalui internet mutlak dibutuhkan tetapi akan membawa implikasi keamanan. Isu keamanan di internet cukup penting karena merupakan infrastruktur jaringan publik yang *insecure* dan *untrustable*, mudah diserang oleh para intruder baik yang amatir maupun yang professional.

Informasi-informasi yang bisa diakses melalui internet memiliki tingkat kerahasiaan yang beragam. Contohnya, dalam bisnis perhotelan, *informasi* penyewaan kamar bisa diakses oleh setiap orang, tetapi ada layanan pemesanan khusus dengan harga dan pembayaran khusus bagi rekanan bisnis, misalnya biro wisata yang memiliki hubungan bisnis *dengan hotel*.

Web service bekerja menggunakan teknologi XML dan SOAP. Dengan kedua teknologi tersebut berarti web service memiliki karakteristik **netral bahasa** dan **netral platform,** tetapi sekaligus memiliki lubang keamanan yang besar [8], [3], [12].

Siddique dalam [13] mengatakan bahwa standar keamanan berbasis XML yang dipublikasikan oleh W3C dan OASIS adalah **XML Signatures, XML Encryption dan Web Services Security (WSS).** Ketiga standard keamanan berbasis XML tersebut, dalam komunikasi SOAP menawarkan fitur keamaman berupa :

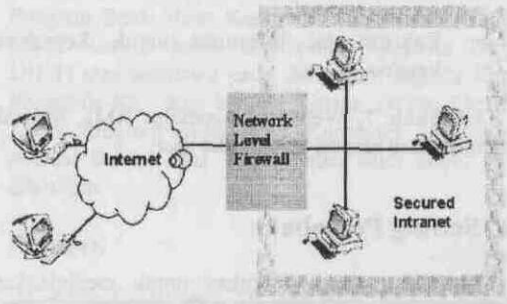
- otentikasi user
- integritas message dan
- kerahasiaan

Salz dalam [14] menuliskan bahwa beberapa kriteria keamanan web service adalah : ,

- **Integritas**, message diterima dalam kondisi asli tidak mengalami perubahan.
- **Rahasia**, message diterima *hanya oleh pihak* yang sah.
- **Otentikasi**, pengirim dapat diketahui secara pasti
- **Otorisasi**, pengirim adalah pemakai yang resmi, *memiliki hak pakai terhadap service tersebut.*

Pengamanan jaringan sering menggunakan firewall. Firewall tradisional bekerja dengan melakukan bloking terhadap port tertentu, nomor IP tertentu atau menutup jalur berdasarkan sifat penggunaannya. Firewall ini, packet-level firewall [3] / network-level firewall [14], tidak mampu membedakan isi message SOAP dan XML yang lewat, sehingga message dapat lewat begitu saja [1]. Network-level firewall digambarkan seperti gambar 2.

Mekanisma yang mungkin untuk melakukan pengamanan web service adalah menggunakan XML Firewall, yaitu modul keamanan yang merupakan gabungan protokol-protokol keamanan yang direkomendasikan oleh W3C dan OASIS[13].



Gambar 2. Network Level Firewall

XML firewall bekerja pada content-level yaitu dengan melakukan pemeriksaan terhadap message yang diterima. XML firewall mampu memeriksa SOAP header dan XML tags, sehingga mampu membedakan message yang *unauthorized* milik pengguna yang tidak sah. [3], [1].

5. Cara Kerja XML Firewall

Bilal Siddiqui, dalam [15] menjelaskan bahwa cara kerja XML Firewall dapat dijelaskan berdasarkan gambar 3.

Langkah 1, User mengirimkan request terhadap suatu service.

Langkah 2, XML firewall menerima request berisi pemanggilan suatu method. Sebelum request diteruskan ke SOAP Server, XML firewall memeriksa dua hal :

- **Otentikasi**, untuk mengetahui identitas pengirim request, XML firewall mengambil salah satu token sekuriti dari request yang datang, kemudian mengolahnya untuk mengetahui siapa pengirimnya.
- **Integritas Message**, untuk meyakinkan bahwa message yang diterima sama dengan message yang dikirim oleh pengirim, XML firewall melakukan validasi pada signaturenya.

Setelah memeriksa otentikasi dan integritas message, XML firewall meneruskan request ke SOAP server.

Langkah 3, XML firewall mengirimkan SOAP request ke SOAP Server.

Langkah 4, SOAP server memanggil web service dan menyiapkan SOAP response.

Langkah 5, SOAP Server mengirimkan balik ke XML firewall.

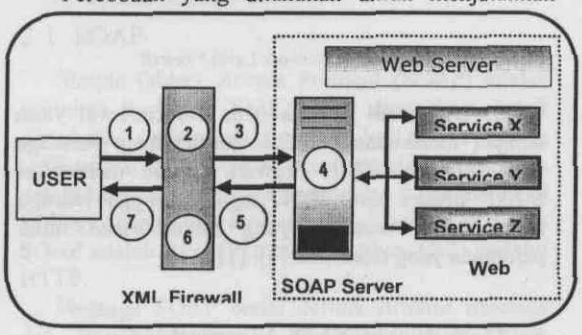
Langkah 6, XML firewall menyiapkan message untuk dikirimkan ke user, didalamnya termasuk :

- Menambahkan token sekuriti kedalam SOAP message sebagai respon dari web server
- Menambahkan XML Digital Signature sebagai tanda dari otentikasi user dan integritas message.
- Enkripsi isi informasi untuk keperluan kerahasiaan.

Langkah 7, Web server melalui XML firewall mengirimkan respon lengkap ke user.

6. Setting Percobaan

Percobaan yang dilakukan untuk menjalankan



Gambar 3. Cara kerja dan tata letak XML Firewall

web service, menggunakan 2 buah komputer, 1 buah sebagai server dan 1 buah yang lain sebagai client. Konfigurasi komputer yang digunakan adalah :

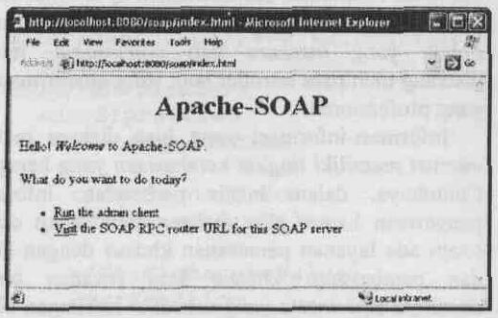
- **Server** : Processor Intel Pentium II 300 MHz, RAM 256Mbyte, NIC, sistem operasi Microsoft Windows 2003 Enterprise Edition,
- **Client** : Processor Intel Pentium II 300 MHz, RAM 256Mbyte, NIC, sistem operasi Microsoft Windows XP Home Edition.

Pada Client dan Server, masing-masing dikonfigurasi untuk membentuk workgroup. Selain sistem operasi, pada kedua komputer perlu diinstal beberapa software untuk *Tool* dan *Engine*. Software yang perlu diinstal, pada sisi server :

- Java™ 2 SDK, Standard Edition 1.4.1
- Java™ 2 SDK, Enterprise Edition 1.3.1
- Java IDE : JCreator 3.0 LE
- SOAP server: Apache SOAP for Java 2.2
- Java Web Server : Apache Jakarta TomCat 3.2.4
- Javabeans™ Activation Framework 1.0.2
- JavaMail API versi 1.2
- Apache Xerces XML Parser 1.2.3
- Text Editor : HTML Kit
- Microsoft Internet Explorer 6.0.

Pada sisi client :

- Java™ 2 SDK, Standard Edition Version 1.4.1
- Java™ 2 SDK, Enterprise Edition Version 1.3.1
- Java IDE : JCreator 3.0 LE
- Microsoft Internet Explorer 6.0



Gambar 4. Tampilan hasil instalasi pada sisi server.

Server yang sudah selesai diinstalasi dapat dicek hasilnya, dengan memanggil URL http://localhost:8080/soap/index.html, jika tidak ada kesalahan maka hasilnya seperti gambar 4, sedang tampilan layar untuk adminstrator seperti gambar 5.

Gambar 5. Tampilan layar adminstrator.

7. Implementasi XML Firewall

```
<?xml version="1.0"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV=
    http://schemas.xmlsoap.org/soap
    /envelope/
  xmlns:ds="http://www.w3.org/2000/
    09/xmldsig#">
  <SOAP-ENV:Header>
    <ds:Signature>
      <ds:SignedInfo>
        <ds:CanonicalizationMethod
          Algorithm="http://www.w3.
          org/2001/10/xml-exc-
          c14n#"/>
        <ds:SignatureMethod
          Algorithm="http://www.w3.
          org/2000/09/xmldsig#rsa-
          sha1"/>
        <ds:Reference
          URI="#GetSpecialDiscounte
          dBookingForPartners">
          <ds:Transforms>
            <ds:Transform
              Algorithm="http://ww
              w.w3.org/2001/10/xml
              -exc-c14n#"/>
          </ds:Transforms>
            <ds:DigestMethod
            Algorithm="http://www.
            w3.org/2000/09/xmldsig
            #sha1"/>
          <ds:DigestValue>
            BIUddkjKKo2...
          </ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
      halHJghyf765....
      </ds:SignatureValue>
      <ds:KeyInfo>
      <ds:KeyName>MyKeyIdentifier<
    /ds:KeyName>
    </ds:KeyInfo>
    </ds:Signature>
  </SOAP-ENV:Header>

  <SOAP-ENV:Body>
    <s:MyServices
    xmlns:s="http://MyURL/service/">
      .....
    <!--Parameters call-->
      .....
    </s:MyServices>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

8. Kesimpulan

Lubang keamanan pada web service dapat diatasi dengan mengimplementasikan XML firewall, yaitu dengan menempatkan token keamanan pada header dalam message SOAP.

Ucapan Terima Kasih

Tulisan ini merupakan bagian dari tesis S2 di Program Studi Ilmu Komputer Universitas Gadjah Mada. Terima kasih disampaikan kepada Dirjen DIKTI atas beasiswa yang diberikan, kepada Ketua Pengelola S2, juga kepada Rektor, Dekan FMIPA, Ketua Program Studi Ilmu Komputer Universitas Ahmad Dahlan atas kesempatan studi lanjut yang diberikan.

Referensi

[1] Kaler, C., (Editor), "WS-Security, Version 1.0. An IBM, Microsoft and VeriSign joint specification", http://www-106.ibm.com/developerworks/ webservices/library/ws-secure/ , 5 April 2002

[2] Newmarch, J., Huang, M., Chua, K.G., "Firewalling Web Service", (http://jan.netcomp.monash.edu.au) , 2002

[3] Smith, R.F., "SOAP/XML Firewalls, Web Services require more protection than traditional firewall offer", http://www.winnetmag.com/Articles/Print.cfm?ArticleIID=39755, September 2003

[4] Gralla, P, "XML Firewall, The Web Service Advisor", http://searchwebservices.techtarget.com/tip/1,289483, sid26_gci855052,00.html., 8 Oktober 2002

[5] Gottschalk, K., Graham, S., Kreger, H., dan Snell, J., 2002, "Introduction to Web services architecture". IBM Systems Journal, Volume 41, Number 2, http://www.research.ibm.com/journal/sj/412/gottschalk.html , 2002

[6] Salz, R., "Securing Web Services," http://webservices.xml.com/pub/a/ws/2003/01/15/ends.html, 15 Januari 2003

[7] Dahiya, R, "Enhanced Security Mechanisms for Multi-party Web Services Applications",

http://nas.cl.uh.edu/yang/teaching/csci5931webSecurity/ ThesisProWS_Rajiv.doc, 2003

[8] Box, D., (DevelopMentor), Ehnebuske, D. (IBM), Kakivaya, G. (Microsoft), Layman, A. (Microsoft), Mendelsohn, N. (Lotus Development Corp.), Nielsen, F. H. (Microsoft), Thatte, S., (Microsoft), Winer D., (UserLand Software, Inc.), "SOAP: Simple Object Access Protocol 1.1. W3C Note" http://www.w3.org/TR/2000/NOTE-SOAP-20000508/, 8 Mei 2000.

[9] Anonim, "What Is SOAP, SOAP Basic 1", http://www.soapuser.com/basics1.html, 2003

[10] Anonim, "What Is SOAP, SOAP Basic 3", http://www.soapuser.com/basics3.html, 2003

[11] Christensen, E. (Microsoft), Curbera, F. (IBM Research), Meredith, G. (Microsoft), Weerawarana, S. (IBM Research), "Web Services Description Language (WSDL) 1.1 W3C Note", http://www.w3.org/TR/2001/NOTE-wsdl-20010315, 15 Maret 2001

[12] Taft D.K. "XML Firewall Aid Services", http://www.eweek.com/article2/0,3959,474122,00.asp , 19 Agustus 2002

[13] Siddiqui, B., "Web Services Security, Part 2" , http://webservices.xml.com/pub/a/ws/2003/04/01/security.html, 1 April 2003

[14] Siddiqui, B., "Web Services Security, Part 1", http://webservices.xml.com/pub/a/ws/2003/03/04/security.html, 4 Maret 2003

[15] Siddiqui, B., "Web Services Security, Part 4" , http://webservices.xml.com/pub/a/ws/2003/04/01/security.html, 22 Juli 2003

Tawar *lulus sarjana S1 Program* Studi Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta tahun *1994, bekerja sebagai staff* pengajar pada program studi Ilmu Komputer, sedang memegang amanah *sebagai Pembantu* Dekan III di Fakultas MIPA Universitas Ahmad Dahlan. Saat ini sedang *menyelesaikan Program* Pasca Sarjana Program Studi Ilmu Komputer, Universitas Gadjah Mada, *Yogyakarta.*

Jazi Eko Istiyanto, menyelesaikan Sarjana Fisika tahun 1986 di Universitas Gadjah Mada, melanjutkan studi di UK dan mendapatkan **Postgraduate Diploma** in Computer Programming and Microprocessor Applications (1987), **M.Sc** pada departemen Computer Science (1988), and **Ph.D** in Electronic Systems Engineering (1995), dari University of Essex. Saat ini menjabat Ketua Program Studi Elektronika dan Instrumentasi pada *Jurusan Fisika* dan juga sebagai staff pengajar pada Program S2 Ilmu Komputer di *Universitas* Gadjah Mada Yogyakarta. Bidang Penelitian yang ditekuni meliputi *information security, web services*, and *embedded systems.* Dia juga aktif sebagai anggota *Himpunan Fisika* Indonesia dan Persatuan Insinyur Indonesia.