Een veilige implementatie van Webservices vraagt om een gedegen afweging van de te nemen beveiligingsmaatregelen waarbij de organisatorische invulling evenveel aandacht vraagt als de technische inrichting. Dankzij de beschikbare toolkits en ontwikkelomgevingen hoeft de technische complexiteit geen struikelblok te zijn voor de inrichting van een goede beveiliging. Dit artikel geeft inzicht in een aantal maatregelen die genomen kunnen worden voor de beveiliging van Webservices. De beschreven technologieën zijn allen gebaseerd op open standaarden.

BEVEILIGING VAN WEBSERVICES

Door: Hans Klunder en Meint Post

1 INLEIDING

Webservices zijn druk bezig met een opmars vanuit het laboratorium naar de echte wereld. Webservices maken het mogelijk applicaties met elkaar te laten samenwerken via een uniform uitwisselingsformaat (SOAP). Er wordt meestal gebruik gemaakt van HTTP (hét standaard protocol van het web) voor het uitwisselen van informatie. SOAP staat voor Simple Object Access Protocol en is een op XML gebaseerde industriestandaard. SOAP zorgt er voor dat verschillende applicatieomgevingen ongeacht hun besturingssysteem en ontwikkeltaal met elkaar kunnen samenwerken. Op deze wijze kan een applicatie op een IBM AIX platform die met J2EE (Java 2 Enterprise Edition) is gemaakt bijvoorbeeld communiceren met een applicatie die met Microsoft .Net gemaakt is zonder dat de ontwikkelaars kennis hoeven te hebben van elkaars platform specifieke implementatie. Dit kan het ontwikkeltraject van gedistribueerde applicaties aanmerkelijk versnellen. Webservices lenen zich goed voor Enterprise Application Integration projecten en B2B toepassingen. Steeds meer leveranciers voorzien hun producten van webservices technologie. Voorbeelden hiervan zijn Microsoft´s .Net, IBM´s WebSphere, Oracle´s 9i Application Server en SAP´s mySAP.

Dit artikel start met een overzicht van de beveiligingsaspecten die een rol spelen bij de beveiliging van webservices. Vervolgens komen de beschikbare maatregelen aan bod gevolgd door aspecten van de implementatie.

2 BEVEILIGINGSASPECTEN

Omdat Webservices een direct koppelvlak vormen met een applicatie, vaak backoffice systemen, is veiligheid een eerste vereiste. De SOAP standaard schrijft niet voor hoe de beveiliging voor Webservices vorm gegeven dient te worden. Er worden wel een aantal mogelijkheden beschreven maar deze zijn niet volledig dekkend.

Bij het beveiligen van Webservices dienen de volgende aspecten minimaal gedekt te worden:
- Beschikbaarheid
- Integriteit
- Vertrouwelijkheid

De hierboven genoemde aspecten worden vaak aangeduid als de BIV-classificatie. De engelstalige variant hiervan is de AIC-classificatie (Availibility, Integrity en Confidentiality). De AIC-classificatie is beschreven in de ISO standaard 17799.

Met behulp van het classificatiesysteem kan een organisatie haar informatiemiddelen waarderen. Op basis van de waardering worden vervolgens de te nemen maatregelen gekozen. Een classificatie loopt vaak van 1 (laag) tot 3 (hoog). Een informatiemiddel, bijvoorbeeld een database met klantgegevens, met de classificatie BIV = 223 moet sterker beschermd worden dan een lager gewaardeerd informatiemiddel zoals de website waarop de interne "Tour de France" pool wordt bijgehouden. Het doel van een classificatiesysteem is om de juiste balans te vinden tussen bedreigingen en te nemen maatregelen zodat de kosten in evenwicht zijn met de risico's. De afweging tussen kosten en risico's is de basis van elke gedegen informatiebeveiliging.

Webservices zijn informatiemiddelen en dus zijn ze te classificeren volgens de BIV-methode. Zoals aangegeven bevat het BIV-classificatiesysteem drie aspecten: Beschikbaarheid, Integriteit en Vertrouwelijkheid.

Met Beschikbaarheid wordt het instandhouden van de continuïteit van de Webservice bedoeld. Niets is vervelender dan dat een kritische applicatie tot stilstand komt omdat er ergens een Webservice niet beschikbaar is. Het is zaak om de beschikbaarheid van alle betrokken Webservices mee te nemen bij de beoordeling van de beschikbaarheid van de applicatie. Hier dienen contractuele afspraken of Service Level Agreements met de leveranciers van Webservices overeen gekomen te worden. Zomaar een gratis Webservice die via Internet aangeboden wordt gebruiken in een kritische rol in applicatie is een onverstandige zaak.

Met Integriteit wordt bedoeld dat het bericht tijdens transport niet aangepast wordt. Het zou buitengewoon kwalijk zijn als een effectenorder plotseling een aankooporder wordt in plaats van de beoogde verkooporder. Integriteitproblemen kunnen ontstaan door storingen in de communicatie en door opzettelijke inbreuken (hacking). Contractueel dient vastgelegd te worden welke partij de verantwoordelijkheid draagt voor eventuele integriteitproblemen bij het gebruik van webservices. Over het algemeen zal de ontvanger impliciet vertrouwen dat alles wat de verzender aanbiedt via de webservice integer is.

Vertrouwelijkheid betekent dat het bericht tijdens transport niet te lezen is. De informatie kan wel afgevangen worden maar is niet begrijpelijk. Niemand zal

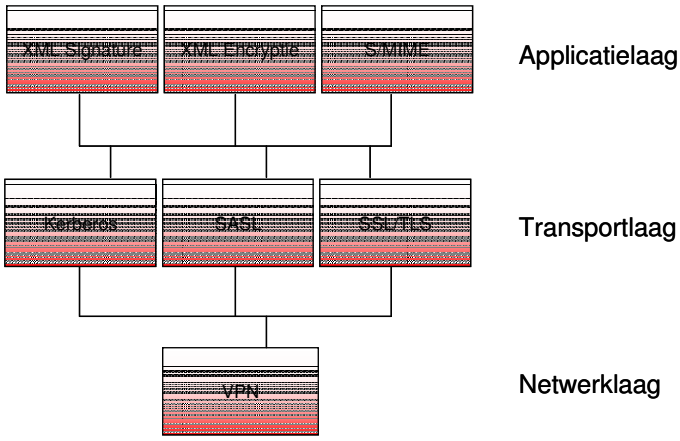
blij zijn als vertrouwelijke medische informatie onderweg van huisarts naar ziekenhuis onderschept en openbaar gemaakt wordt. Contractueel dient vastgelegd te worden welke partij aansprakelijk is als informatie toch openbaar raakt naar aanleiding van het gebruik van een webservice.

Natuurlijk is de beveiliging van een webservice niet alleen afhankelijk van bescherming van de service zelf. Het platform waar de webservices op aangeboden worden, de fysieke omgeving waar het geheel in gehuisvest wordt en de inzet van personeel spelen een minstens zo belangrijke rol. Ook Beschikbaarheid is meestal het resultaat van een combinatie van organisatorische maatregelen en platformkeuze. Omdat al deze aspecten niet specifiek zijn voor webservices worden ze binnen dit artikel buiten beschouwing gelaten.

Als een Webservice eenmaal geclassificeerd is wordt het mogelijk om de te bepalen welke maatregelen getroffen dienen te worden. Hiervoor wordt het instrument van de risicoanalyse gebruikt. De risicoanalyse is een gestructureerde methode om te bepalen welke bedreigingen er zijn en wat de passende maatregelen zijn.

3 MAATREGELEN

Om webservices te beschermen kunnen er op verschillende niveaus maatregelen genomen worden. Voor de beveiliging van webservices kan er gewerkt worden met een drie lagenmodel (zie Figuur 1).



Figuur 1: lagenmodel

Het lagenmodel is gebaseerd op drie lagen:
- Applicatielaag;
- Transportlaag;
- Netwerklaag;

De applicatielaag is de laag waarin de berichten vormgegeven worden of geïnterpreteerd worden. Aanpassingen in deze laag betekenen aanpassingen in de gebruikte applicaties. De transportlaag houdt zich bezig met het uitwisselen van berichten. Voorbeelden hiervan zijn HTTP, SMTP en leverancierspecifieke oplossingen zoals MQSeries van IBM. De netwerklaag transporteert informatie over fysieke netwerken ongeacht het soort informatie dat overgedragen wordt.

Maatregelen in de applicatielaag kunnen onafhankelijk van de gebruikte transportlaag toegepast worden. Er kunnen voorzieningen getroffen worden voor integriteit en vertrouwelijkheid. Voor Integriteit zijn XML Signature en S/MIME aantrekkelijk. Deze standaarden maken gebruik van digitale handtekeningen voor de bescherming van de integriteit van het bericht. XML Signature is open standaard die bescherming van XML berichten biedt. XML Signature werkt door het toevoegen van specifieke elementen aan een XML bericht. De digitale handtekening kan geplaatst worden op een volledig XML bericht maar ook op individuele elementen uit het bericht. Een XML bericht met een digitale handtekening ziet er als volgt uit (ingekort):

```
<?xml version='1.0'?>
<Document>
…
        <Signature Id="HelloWorldSignature" xm-
lns="http://www.w3.org/2000/09/xmldsig#">
                <SignedInfo>
                <mydata>
                Hello world !
                </mydata>
                </SignedInfo>
                …
                <SignatureVa-
lue>XYZ</SignatureValue>
                …
        </Signature>
</Document>
```

De toegevoegde elementen bevatten informatie over wie het bericht getekend heeft, welke algoritmes gebruikt zijn voor het tekenen (bijvoorbeeld SHA1/RSA) en de digitale handtekening zelf (de SignatureValue in het voorbeeld).

S/MIME is een wat oudere standaard die voortkomt uit de email wereld. Vrijwel alle moderne e-mail pakketten ondersteunen S/MIME. Het maakt voor S/MIME niet uit of het te tekenen bericht XML bevat, het mag ook binaire informatie zijn. S/MIME voegt aan het bericht een bijlage toe (body part) waarin de digitale handtekening is opgeslagen.

Zowel XML Signature als S/MIME kunnen gebruikt worden voor het rechtsgeldig bewijzen van transacties conform de Europese Richtlijn "Raamwerk voor Digitale Handtekeningen". XML Signature is aantrekkelijk voor grotere berichten waarbij een of enkele elementen beveiligd dienen te zijn. S/MIME is aantrekkelijk bij kleine(re) berichten die in hun geheel beveiligd moeten zijn.

XML Encryptie biedt voorzieningen voor het beschermen van Vertrouwelijkheid van berichten in de applicatielaag. Ook S/MIME kan deze bescherming bieden. In

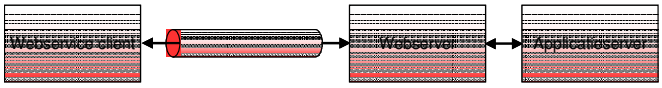
het volgende voorbeeld van een XML bericht met encryptie is alleen het creditcard nummer versleuteld:

```
<?xml version='1.0'?>
<PaymentInfo
xmlns='http://example.org/paymentv2'>
<Name>John Smith<Name/>
<CardNumber>
        <EncryptedData xm-
lns='http://www.w3.org/2001/04/xmlenc#'
        Ty-
pe='http://www.w3.org/2001/04/xmlenc#Content'>
                <Cipher-
Data><CipherValue>A23B45C56</CipherValue></Cip
herData>
      </EncryptedData>
</CardNumber>
</PaymentInfo>
```

Net als XML Signature is XML Encryptie aantrekkelijk voor grotere berichten waarbij een of enkele elementen beveiligd dienen te zijn. S/MIME is wederom aantrekkelijk bij kleine(re) berichten die in hun geheel beveiligd moeten zijn.
Onder de applicatielaag bevindt zich de transportlaag. De transportlaag verzorgt de uitwisseling van berichten. Ook in de transportlaag zijn er diverse mogelijkheden om informatie te beschermen. Het voordeel van maatregelen die getroffen worden in de transportlaag is dat de bovenliggende applicatie niet aangepast hoeft te worden. In het algemeen is het voldoende om alleen de configuratie van het platform aan te passen. Oplossingen op het niveau van de transportlaag bieden meestal maatregelen voor de bescherming van zowel Integriteit als Vertrouwelijkheid. De bekendste oplossingen op dit niveau zijn SSL/TLS, Kerberos en SASL. SSL/TLS is hét beveiligingsprotocol over het Internet. Hoewel SSL/TLS voornamelijk bekend is voor het beschermen van HTTP verkeer (websites) kan het ook gebruikt worden voor een reeks van andere protocollen (SMTP, FTP, MQSeries, TCP/IP sockets etc...) die gebruikt kunnen worden als transportprotocol voor webservices. Kerberos is een standaard beveiligingsprotocol dat afkomstig is uit de Unix wereld maar inmiddels ook het standaardprotocol voor de beveiliging van Microsoft systemen is (Windows 2000 en hoger). SASL is een wat minder bekend protocol dat veelvuldig wordt toegepast in e-mail beveiliging, het is echter niet beperkt tot alleen e-mail.

Een interessante nieuwkomer die speciaal bedoeld is voor het beveiligen van webservices is het voorstel genaamd SOAP DSIG. SOAP DSIG is een toepassing van XML Signature als onderdeel van het SOAP protocol.

Alle genoemde protocollen zijn in staat om authenticatie van afnemer en aanbieder van webservices op een betrouwbare wijze uit te voeren. Bij het implementeren van beveiligingsmaatregelen voor webservices op het niveau van de transportlaag dient men rekening te houden met performance issues omdat de afhandeling op dit niveau meer capaciteit vraagt dan op applicatief niveau. Daarnaast bieden maatregelen op het niveau van de transportlaag geen end-to-end bescherming (zie Figuur 2). Als een Webservice client via een webserver gebruik maakt van een Webservice op een applicatieserver dan is alleen de verbinding tussen Webservice client en Webserver beschermd.



Figuur 2: geen end-to-end bescherming

Door de wijdverbreide beschikbaarheid van alle drie de protocollen is het erg aantrekkelijk om ze te gebruiken voor de beveiliging van Webservices. Het meest gebruikte protocol is SSL/TLS omdat dit ook in heterogene omgevingen weinig configuratieinspanning vereist.

Als laatste van de drie lagen uit het model komt de Netwerklaag aan bod. Net zoals bij de Transportlaag biedt de Netwerklaag ook vaak maatregelen die zowel Integriteit als Vertrouwelijkheid beschermen. De bekendste maatregel in de Netwerklaag is het Virtual Private Network (VPN). Een VPN is een beveiligde tunnel die aangelegd wordt over een netwerk. Een voorbeeld hiervan zijn twee netwerk routers die samen een tunnel opzetten. Al het verkeer tussen deze twee routers loopt nu door de tunnel en is daarmee beschermd. Een VPN leent zich uitstekend voor de bescherming van informatie tijdens transport over Lan, Wan of Internet. Het heeft de minste impact op de rest van het netwerk en de betrokken applicaties maar is relatief complex qua inrichting en beheer. Een VPN is over het algemeen niet geschikt voor authenticatie van afnemers op applicatieniveau.

Een VPN is aantrekkelijk onder een of meer van de volgende voorwaarden:
- Bekende partners die regelmatig met elkaar communiceren;
- Lan-naar-Lan koppelingen over een publiek netwerk;
- Grote hoeveelheden data die beschermd getransporteerd dienen te worden;
- Er wordt gebruik gemaakt van een protocol dat niet beschermd kan worden met standaard webtechnologie zoals SSL, Kerberos of SASL. Voorbeelden hiervan zijn SNA, IPX/SPX of AppleTalk.

4 IMPLEMENTATIE

Gelukkig hoeft de gemiddelde ontwikkelaar zich niet bezig te houden met de details van de bovengenoemde maatregelen. Er zijn diverse toolkits en ontwikkelomgevingen beschikbaar om de hierboven beschreven maatregelen toe te passen. Het toepassen van een maatregel vereist echter meer dan alleen een technisch kunstje. Het is belangrijk dat de organisatie kennis en begrip heeft van de toegepaste technologie. Goede maatrege-

len die verkeerd toegepast worden kunnen meer schade aanrichten dan geen maatregelen.

5 CONCLUSIE

Een veilige implementatie van Webservices vraagt om een gedegen afweging van de te nemen beveiligingsmaatregelen waarbij de organisatorische invulling evenveel aandacht vraagt als de technische inrichting. Dankzij de beschikbare toolkits en ontwikkelomgevingen hoeft de technische complexiteit geen struikelblok te zijn voor de inrichting van een goede beveiliging.

De auteurs

Mr. M.E. (Meint) Post CISSP en Ing. J.A.A. (Hans) Klunder CISSP zijn ICT Architect en gespecialiseerd in het ontwerp en realisatie van ICT beveiliging. Zij adviseren grotere en kleinere organisaties over de wijze waarop ICT-beveiliging (technisch en organisatorisch) een onderscheidende factor kan zijn in de markt en in de bedrijfsvoering. Ze zijn bereikbaar onder info@more-secure.nl.

Contact

More-Secure BV
Da Costalaan 14
3767GH Soest
T: +31 (0)6 5357 9338
F: +31 (0)35 524 7587
E: info@more-secure.nl
W: www.more-secure.nl