www.bt-magazin.de 1.2012 Heft 8 Deutschland: € 14,90 Österreich: € 16,80 Schweiz: sFr 29,80 Luxemburg: € 16,90

Business Technology

Business Technology

Architektur & Management Magazin

BT-GEPRÜFT

Expertenwissen für IT-Architekten, Projektleiter und Berater

Qualität

Mythos Qualitätsmanagement

Höhere Qualität, bessere Software

„Funktioniert“ ist nicht genug

4 191858 414906 01

Grundlagen des Testens von SOA Security

Sicher testen mit SOA

Da serviceorientierte Architekturen (SOA), basierend auf Web Services, zunehmend die Grundlage jeder modernen IT-Infrastruktur bilden, sind proaktive Tests von Web Services, die das Thema Security berücksichtigen, entscheidend für die IT-Sicherheit im Allgemeinen. Dieser Artikel beschreibt die Grundlagen des Testens der Sicherheit von Web Services und gliedert das Thema in die Überprüfung der Funktion, Performance, Interoperabilität und Angreifbarkeit.

Sicherheitstests steigern das Vertrauen in Services dadurch, dass die Prinzipien einer unfälschbaren Identität, Privatsphäre und Integrität geprüft und Bedrohungen wie Denial of Service, Malware und Datenlecks von den SOA-Implementierungen ferngehalten werden. Funktionsreiche Internetapplikationen, Service-APIs und Virtualisierungen sowie Cloud Services bieten eine umfassende Integration von Daten, die die Nutzung der Informationen in Echtzeit erlaubt. Diese großartigen Möglichkeiten, die zweifellos den Wert von Businessanwendungen steigern, haben ihren Preis: die Sicherheit der Services. Dieser Artikel befasst sich mit dem Testen von SOA Security. Die Vertrauenswürdigkeit eines Service ist ein entscheidender Faktor dafür, ob ein potenzieller Konsument ein Serviceangebot nutzt oder nicht. Interessanterweise vernachlässigen viele Anbieter von Web Services diese Tatsache. Sie gehen davon aus, dass ein reichhaltiger Funktionsumfang für die meisten Verbraucher gut genug ist. Doch die Erfahrung der Autoren zeigt, dass oftmals nicht funktionale Anforderungen den Unterschied zwischen erfolgreichen Serviceangeboten und ergebnislosen, akademischen Versuchen ausmachen, SOA in einer Organisation durchzusetzen. Aufgrund der komplexen Natur von SOA, die viele Systeme, Protokolle, Datentypen, Identitätstoken, Verschlüsselungsmechanismen und Signaturtechniken beinhaltet, erfordert das Testen der Sicherheit in einer SOA erhebliche Disziplin. Komplexe Geschäftsfunktionen, die durch Web Services zugänglich gemacht werden können, verarbeiten häufig sensible Informationen wie Kundendaten, Bestellungen, Steuererklärungen, Finanzberichte oder Gesundheitsscans. Aus diesem Grund sind Tests, die nachweisen, dass bei der Übertragung und Verarbeitung solcher komplexen Strukturen die Spielregeln im Umgang mit Benutzeridentität, Privatsphäre und Integrität eingehalten werden, essenziell für den Aufbau einer sicheren SOA.

FUNKTIONSÜBERPRÜFUNG

Die Funktionsüberprüfung ist der erste Eckpfeiler von SOA-Security-Tests. Hier geht es darum zu prüfen, ob ein Service geforderte Richtlinien überhaupt erfüllt. Ist beispielsweise definiert, dass ein Service nur Nachrichten verarbeiten darf, die verschlüsselt übertragen wurden, dann ist es Aufgabe des Testers, genau dieses Verhalten nachzuweisen. Nachdem solche grundlegenden Tests durchgeführt wurden, entwerfen Tester Regressionstestfälle für die Automatisierung des Tests. Dabei sind in der Regel folgende Punkte zu beachten:

1. **Transportprotokolle**: Services sollten unabhängig vom Transportprotokoll funktionieren. Die meisten Internetdienste vertrauen in puncto Kommunikation auf HTTP oder HTTPS. Innerhalb eines Unternehmens sind jedoch JMS, IBM MQ Series, Tibco EMS und FTP weitere beliebte Transportprotokolle. Ihre Vielfalt verlangt, dass die Testumgebung in der Lage ist, Nachrichten mit diesen Protokollen sowohl zu senden als auch zu empfangen. Bei der Verwendung von SSL wird zudem das Management einer Public-Key-Infrastruktur erforderlich.
2. **Identity-Token**: Die meisten Services verlangen die Authentifizierung und Autorisierung des Benutzers, bevor ein Aufruf überhaupt angenommen und eine Antwort zurückgegeben wird. Die Identität des Benutzers kann über die verschiedensten Kanäle übertragen werden:

   - HTTP Basic Authentication
   - Cookie-based Authentication
   - HTTP x.509 Mutual Authentication
   - SAML, WS-Username, WS-X.509
   - Ad-hoc-Content in der Nachricht (Header, Nachricht, Anlage)

3. **Datenschutz und Integrität**: SOA Security stellt durch Verschlüsselung den Datenschutz und per Signaturverfahren die Integrität von Daten sicher. Dieser Schutz ist sowohl während des Transports als auch für den Fall der Speicherung gegeben. Die Verbindung des Schutzes der Transportstrecke und der Nachrichteninhalte ermöglicht dem Unternehmen eine gute Kontrolle bei der Implementierung seiner Sicherheitsrichtlinien. Auf der anderen Seite entsteht mit dieser Flexibilität ein hoher Anspruch an das QS- und Testteam, die verschiedene Variationen der in der Praxis eingesetzten Standards in Tests widerspiegeln müssen. SOA-Implementierungen erfordern die Prüfung folgender Securityelemente:

   - HTTP über SSL
   - SOAP/XML-Verschlüsselung
   - SOAP/XML-Signaturen
   - MTOM und SOAP inklusive Anhänge

Funktionale Sicherheitstests sind kein triviales Unterfangen. Ein gutes Know-how ist nötig, um die große Vielfalt an Protokollen, Identity-Tokentypen und Sicherheitsrichtlinien im Test zu beherrschen. Man beginnt mit der Erstellung von einzelnen Testfällen, um im

nächsten Schritt zur Automatisierung fortzuschreiten. Durch die Automatisierung können Tester notwendige Regressionstests ohne großen Aufwand durchführen, um sicherzustellen, dass sich ein neues Release eines Service wie erwartet verhält. In der Praxis erweisen sich professionelle Werkzeuge als unverzichtbar, um diese Regressionstest zu entwickeln und detaillierte Berichte ausgeführter Tests zu generieren. Nur so können Entwickler Produktivität und Qualität auf einem hohen Niveau halten.

PERFORMANCE

Der zweite Eckpfeiler des Testens von SOA Security sind die Performancetests. Hier sollen die Tester die Skalierbarkeit und Robustheit der Services testen. Typische Fragen drehen sich beispielsweise darum, ob ein Server eine bestimmte Anzahl an Nachrichten pro Sekunde verschlüsseln kann, und wie sich die Verschlüsselung auf die Antwortzeiten des Endbenutzers auswirkt. Dabei ist es wichtig, Messwerte über die Reaktions- und Latenzzeiten sowie Durchsatzprofile der getesteten Services zu ermitteln. Außerdem ist es wichtig, durch den Beschuss der Services mit verschiedensten Nachrichten, die sich in Größe und Inhalt unterscheiden, das System auf dessen Skalierbarkeit zu überprüfen. Idealerweise imitieren solche Lastprofile das echte Benutzerverhalten. Hierbei haben die folgenden Bereiche Einfluss auf die Performance der SOA Security.

Erzeugen von eindeutigen Nachrichten: Performancetests variieren Nachrichtengrößen, Lastprofile und Zugriffsmuster und überprüfen jeweils den erreichten Durchsatz, die Geschwindigkeit und Latenzzeit. Typischerweise wird hierbei der Zielservice mit zahlreichen Nachrichten überschwemmt und man erhält somit die Anzahl der Transaktionen pro Sekunde (TPS), die ein System verarbeiten kann. Dabei reichen statische XML-Nachrichten häufig nicht aus, da das wiederholte Senden derselben Nachrichten von entsprechend geschützten Servern als Replay-Attacke gewertet und abgeblockt werden würde. Der Tester muss sicherstellen, dass jede SOAP-Nachricht einen dynamisch erzeugten Zeitstempel und einen Security Header besitzt. Die entsprechenden Spezifikationen erfordern eindeutige Signaturen für alle Nachrichten. Der Tester hat anschließend auch die Aufgabe zu überprüfen, ob die Transaktionen, die SOA Security erfordern, ohne Fehler abgearbeitet wurden. Hierbei muss man sich klarmachen, dass man für einen relevanten Performancetest häufig viele tausend Nachrichten benötigt, die man von Hand nicht erzeugen kann.

Skalieren der Clients und Nachrichten: Eine SOA umfasst viele unterschiedliche Anwendungsfälle. In der System-zu-System-Kommunikation werden Transaktionen gewöhnlich über feste Connections übertragen. Im Gegensatz dazu werden beim Benutzer-zu-System-Modell viele kurzlebige, gleichzeitige Verbindungen generiert. Bevor ein Service in die Produktion überführt werden darf, sind außerdem Tests notwendig, in denen die Anzahl der Konsumenten variiert wird. Ähnlich liegt die Sache bei Services, die mit verschiedenen Nachrichtengrößen arbeiten müssen. Ein System, das Steuererklärungen verarbeitet, muss zum Beispiel sowohl mit ein paar Kilobyte aus kleinen Unternehmen als auch mit mehreren Gigabyte von globalen Konzernen umgehen können. Ein Test solcher Systeme umfasst die Anwendung der Sicherheitsvorkehrungen wie Verschlüsselung und Signaturen auf verschiedenen Nachrichtengrößen und die Messung von Geschwindigkeit, Latenz und Durchsatz. Für den Tester ist es mithin wichtig, auf einfache Weise die Anzahl der Testclients und Nachrichtengrößen variieren zu können, um verschiedene Lastprofile erstellen zu können.

Security-Operationen und -schlüssel: Während einiger Tests muss der Tester auch die Performanceeinbußen aufgrund kryptografischer Operationen überprüfen. Viele Security-Operationen erfordern Schlüsselpaare (PKI). Um Daten zu verschlüsseln, wird der öffentliche Schlüssel verwendet. Der private Schlüssel dient der Entschlüsselung. Für die Integrität hat der private Schlüssel die Funktion, das Signieren der Nachrichten zu ermöglichen und der öffentliche Schlüssel dient zur Prüfung der Signatur. Die Rechenoperationen mit privaten Schlüsseln sind rechenintensiver als die Operationen auf öffentlichen und haben somit einen größeren Einfluss auf die Performance. Ebenso hat die Schlüsselgröße erhebliche Auswirkungen auf die Serviceperformance. Das gilt besonders bei Operationen mit privaten Schlüsseln. Um eine hohe Sicherheit zu erreichen, werden Schlüsselgrößen bis zu 4096 Bits benötigt, mindestens jedoch 1024 Bits. Schlüssel unterhalb dieser Größe sind in der Regel zu schwach und werden selten eingesetzt. Da die Verarbeitung von privaten Schlüsseln so rechenintensiv ist, wird hierfür häufig dedizierte und spezialisierte Hardware eingesetzt. Hier ist es auch die Aufgabe des SOA-Security-Testers zu verstehen, welche Schlüsselgrößen für die Umsetzung der Unternehmensrichtlinien erfordert werden.

Leistungstests für die SOA-Sicherheit unterscheiden sich deutlich von Performancetests für Webseiten. Ein gängiger Fehler ist es, bestehende Tools zum Test von Webanwendungen für den Test von Web Service zu verwenden. Hierbei werden statische WS-Security-Nachrichten unter Verwendung von „Cut and Paste“-Mechanismen erzeugt und durch ein Lasttest-

tool ausgewertet. Das führt zu falschen Ergebnissen. Die Vielzahl an relevanten Details der WS-Security kann abschreckend wirken. Deren Verständnis ist aber unverzichtbar für die richtige Nutzung der richtigen Werkzeuge und den Aufbau der Performancetestumgebung, die relevante Ergebnisse liefert. In der Praxis hat sich die Nutzung geeigneter Tools als entscheidend herausgestellt. Die meisten SOA-Testtools unterstützen die Tester insbesondere dabei, geeignete Lastprofile zu erzeugen.

INTEROPERABILITY

Der dritte Eckpfeiler des SOA-Security-Testens ist der Interoperabilitätstest. Eines der Versprechen von SOA ist, dass es eine einfache Integration zwischen Anwendungen und Systemen ermöglicht. Hierbei ermöglichen die SOA-Anwendungen einen von Programmiersprachen wie Java, .NET oder PHP und Betriebssystemen wie Windows, Linux oder Solaris unabhängigen Austausch von Servicedefinitionen zur Entwicklungszeit und entsprechenden Nachrichtenaustausch zur Laufzeit. Sollten nun Sicherheitsmechanismen aktiviert sein, wird der Interoperabilitätstest in mehreren Bereichen schwieriger:

- **Transport Protocol Security**: HTTP inklusive SSL werden in den meisten SOA-Implementierungen als De-facto-Transportprotokoll verwendet. Während eine sichere Transportverbindung zwischen dem Client und einem Server hergestellt wird, werden kryptografische Algorithmen sowohl vom Client als auch vom Server ausgehandelt. Für verschiedene Sicherheitsanforderungen existieren stärkere und schwächere Algorithmen, die naturgemäß eine unterschiedliche Performance haben. Für den Tester gilt es nun zu überprüfen, wie sich der Client beim Aufbau einer sicheren Transportverbindung verhält und welche Algorithmen verwendet werden. In diesem Testlauf ist der erste und wichtigste Schritt, die SSL-Verbindung eines Service in einer Vielzahl von Szenarien zu überprüfen.
- **Interoperabilität der Benutzeridentitäten**: Wie schon im Abschnitt der funktionalen Tests hervorgehoben, können durch einen Service verschiedene Identity-Token für die Clientauthentifizierung und -autorisierung akzeptiert werden. Diese Tokentypen können auch in unterschiedliche Versionen auftreten. Beliebte Beispiele hierfür sind SAML 1.1 und SAML 2.0. Tester sind hierbei verantwortlich für die Überprüfung, ob ein Service mit jedem unterstützen Tokentyp gleichermaßen funktioniert. Auf der Konsumentenseite müssen dabei verschiedene Programmiersprachen und Betriebssysteme variiert werden. Außerdem ist sicherzustellen, dass nicht akzeptierte Tokentypen, die der Service nicht unterstützt, mit einer entsprechenden Fehlermeldung abgelehnt werden.
- **Interoperability bei Datenschutz und Integrität**: Ähnlich wie beim funktionalen Securitytesten können Operationen zum Datenschutz und zur Integrität verschiedene Probleme mit der Interoperabilität aufwerfen. Auch hier spielen die Security-Token, öffentliche Schlüssel (für die Verschlüsselung) und private Schlüssel (zum Entschlüsseln) eine wichtige Rolle. Zusätzlich zu den Token sind sensible Inhalte in XML- oder SOAP-Nachrichten verschlüsselt. Die Aufgabe des SOA-Security-Testers ist hierbei sicherzustellen, dass nur die richtigen Schlüsselpaare eine Entschlüsselung ermöglichen. Für die Standards beim Signieren werden ebenfalls private Schlüssel (für das Signieren) und öffentliche Schlüssel (zum Verifizieren) verwendet. Zusätzlich zu den grundlegenden Operationen erlauben die Standards, viele Spezialfälle abzudecken, zum Beispiel Teile einer Nachricht bei der Signatur auszuklammern. Diese Flexibilität ist zwar beeindruckend und in der Praxis überaus nützlich, stellt aber eine erhebliche Belastung für die Tests innerhalb einer SOA-Implementierung dar.

Bei der Verwendung eines Service müssen die Konsumenten sowohl zur Entwurfszeit als auch zur Laufzeit Konventionen zur Interoperabilität einhalten. Entwickler und Tester sollten umfassende Interoperabilitätstests durchführen und mögliche Probleme dokumentieren. Erstellt man bestehend aus vielen Einzeltests für sein Umfeld eine umfassende Testsuite, die alle wichtigen Eigenschaften der Interoperabilität abdeckt, wird die nahtlose Zusammenarbeit aller Services auch nach fachlichen und technischen Änderungen und Erweiterungen sichergestellt. Frühzeitige, mit Tools automatisierte Interoperabilitätstests sorgen dafür, dass das Wissen um Interoperabilitätseigenschaften operationalisiert und personenunabhängig wird.

ANGREIFBARKEIT

Der vierte Eckpfeiler des SOA-Security-Testens sind Tests der Angreifbarkeit. Mithilfe von SOA werden unternehmensinterne IT-Ressourcen externen Nutzern bereitgestellt. SOA erweitert damit die „Angriffsfläche“ der Unternehmens-IT. Da nun Services öffentlich zugänglich sind, sind auch Details zu Datentypen, Protokollen, Input- und Outputnachrichten öffentlich zugänglich. Diese Informationen bieten die Grundlage für viele gängige Angriffe:

- **Injection und Data Excavation**: SQL Injection ist eine bekannte Angriffstechnik, die häufig angewandt

wird, um sich unerlaubten Zugriff auf Internetdatenbanken zu erschleichen. Die Angriffstechnik ermöglicht es, abhängig von der Abfrage, die gesamte Datenbank zu löschen oder sensible Inhalte, wie Benutzernamen und Passwörter, aus der Datenbank zu stehlen. Um solche Abfragen zu verhindern, benötigt man eine Vorverarbeitung der Eingabefelder, bevor diese an die Datenbank weitergeleitet werden. Mit der zunehmenden Verbreitung der SOA-Implementierungen verstärken sich auch die Bedrohungen, da Services, die solche Schwachstellen aufweisen, gegebenenfalls mehrfach in verschiedenen Geschäftsprozessen eingesetzt werden. Hierbei dienen nun SOAP-Nachrichten als Angriffsfläche der SQL Injections. Die Aufgabe eines SOA-Security-Testers ist es hierbei, als Angreifer zu agieren und Angriffsszenarien als Testfälle zu definieren, um die Reaktion der Services auf diese Angriffe zu prüfen.

- **Viren und Malware:** SOA bietet die Möglichkeiten, jede Art des Inhalts als Anhang an die XML- oder SOAP-Nachricht zu senden. Unternehmen nutzen diese Möglichkeit, um komplexe digitale Daten auszutauschen. Malware und Viren können über diese Kanäle in Unternehmen eingeschleust werden. SOA-Security-Tester sollten zu Testzwecken „gutartige" Malware und Viren versenden, um sicherzustellen, dass die Zielservices diese infizierten Anfragen ablehnen.
- **Ressourcenverbrauch:** Informationen, die in Servicedefinitionen zur Verfügung gestellt werden, können Angriffsvektoren wie Buffer-Overflows und Endlosschleifen ermöglichen. Diese können zum Aufbau von Denial-of-Service-Angriffen genutzt werden und damit zu Betriebsstörungen führen. Tester sollten eine Reihe von Testfällen konstruieren, um diese Schwachstellen von Service offenzulegen.

Tester müssen sicherstellen, dass Schwachstellen wie Buffer-Overflows tief geschachtelte XML-Strukturen, rekursive Nutzlasten, Schema Poisoning oder Malware keinen Einfluss auf die Services haben. Sie benötigen die Fähigkeiten, Services zu untersuchen und zu bewerten, ob Schwachstellen existieren. Die Prüfung, ob Services angreifbar sind, endet nicht mit der Produktionseinführung. Viel mehr handelt es sich um eine permanente Aufgabe.

SCHLUSSFOLGERUNGEN

SOA versetzt Unternehmen in die Lage, sehr effizient Daten in Echtzeit auszutauschen. Der deutliche Anstieg der entsprechenden Nutzungszahlen sowie neue Cloud-basierte Dienste verbessern Kosten und Nutzen vieler IT-Anwendungen. Nun steht die Professionalisierung dieser Technologien auf der Agenda vieler Unternehmen. Wichtige Projekte, die Services bereitstellen oder konsumieren, wenden zunehmend den Blick auf die Sicherheit dieser Services und weisen grundlegende Sicherheitseigenschaften in Tests nach. Unternehmen erkennen, dass das Testen von SOA Security anspruchsvolle Aspekte besitzt, die über das Testen einer einfachen Webseite hinausgehen. Der Aufbau eines kompetenten SOA-Security-Teams, die Auswahl der SOA-Testing-Tools sowie der Aufbau von Unternehmensprozessen, die den Test von Services sicherstellen, sind entscheidend für eine nachhaltig erfolgreiche SOA-Implementierung.

Mamoon Yunus

ist CEO von Crosscheck Networks, einem führenden Technologieanbieter für Cloud-und Web-Service-Infrastrukturen. Als SOA-Pionier und Gründer von Forum Systems hat er wichtige Techniken für XML Appliances patentieren lassen. Er besitzt zwei Abschlüsse vom MIT. InfoWorld hat ihn 2004 als einen von vier „Up and coming CTOs to watch" ausgezeichnet.

Dr. Dirk Krafzig

ist Gründer von SOAPARK. Als Sprecher auf Konferenzen und Autor von Artikeln und Büchern gilt Dr. Krafzig als ein Protagonist der serviceorientierten Architektur (SOA) und hat maßgeblich zu der Begriffsbildung in diesem Bereich beigetragen. Insbesondere die SOA-Fallstudien mit der Deutschen Post, Credit Suisse, Halifax Bank of Scotland und Winterthur Versicherung in seinem Bestseller „Enterprise SOA" haben viel Aufmerksamkeit auf sich gezogen. Derzeit arbeitet Dr. Krafzig in einem strategischen SOA-Programm bei einem Mobilfunkanbieter an dem Thema Security.